Bootstrap 3.3.5

Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos

Maurício Samy Silva
www.maujor.com

novatec

Bootstrap é um poderoso framework front-end voltado para o desenvolvimento rápido e fácil de sites e aplicações web responsivos e alinhados com a filosofia mobile-first. É indicado para dispositivos e projetos de todos os tamanhos e destinado a desenvolvedores com qualquer nível de conhecimento,

Assuntos tratados no livro:

- Sistema de grid e sua aplicação na construção de layouts dos mais simples aos mais complexos.
- Estilização-padrão para elementos HTML tipográficos de marcação de códigos, tabelas, formulários, botões, imagens e ícones.
- Criação e estilização de elementos de interface gráfica, como barras de navegação, janelas modais, navegação em abas, menus dropdown, paginação, barras de progresso, painéis, agrupamento e listas.
- Integração das funcionalidades do framework com a biblioteca jQuery para obtenção de tooltip, scrollspy, popover, alertas, efeito carrossel, affix e colapse.

O último capítulo do livro mostra como criar, passo a passo, a marcação da homepage de um site a partir da imagem da página usando o grid e as funcionalidades do Bootstrap.

Os assuntos e as funcionalidades mostrados no livro, quando julgado conveniente, são ilustrados com um exemplo prático que se encontra disponível para consulta online em *http://livrosdomaujor.com.br/bootstrap3/codigos.html*.

Neste livro você aprenderá o essencial para obter o máximo proveito dos recursos deste incrível framework e será capaz de criar aplicações web com qualidade profissional.

Bootstrap 3.3.5

Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos

Maurício Samy Silva

Novatec

Dedico este livro a Mark Otto e Jacob Thornton,

criadores do framework Bootstrap.

Sumário

Agradecimentos

Agradeço a Deus por ter me dado forças, disposição e motivação para escrever este livro.

Meu maior agradecimento é a você, leitor, por interessar-se em aprender as técnicas de construção de layout com uso do framework Bootstrap e honrar-me com a leitura deste livro.

Isenção de responsabilidade

Todos os esforços foram feitos na elaboração deste livro para assegurar o fornecimento de informações as mais precisas, completas e exatas. Contudo as informações aqui contidas são fornecidas "como estão" e sem nenhuma garantia, seja expressa, seja implícita. O autor, a editora, os distribuidores e qualquer entidade envolvida direta ou indiretamente na sua comercialização não assumirão nenhuma responsabilidade por qualquer prejuízo ou dano, direto ou indireto, consequente às informações contidas neste livro.

Sobre o autor

Maurício Samy Silva é graduado em Engenharia Civil pelo Instituto Militar de Engenharia (IME). Sua experiência com desenvolvimento de sites teve início em 1999, motivada pela necessidade de criar um site para um clube de pesca amadora do qual ele era o presidente e uma espécie de faz-tudo.

Em 2002, ele teve seu primeiro contato com o site do W3C e começou a pesquisar e a estudar as Web Standards, tendo como fonte de consulta o material publicado na internet em língua inglesa.

No segundo semestre de 2003, estimulado pela completa falta de material de consulta gratuita na internet, ele decidiu lançar o Site do Maujor, o qual é nacionalmente conhecido, hospedado em *http://www.maujor.com/*. A proposta inicial do site era divulgar a cascading style sheet (CSS), ou folha de estilo em cascata, com base no compartilhamento de suas experiências com tal técnica. Com a pronta aceitação e o sucesso crescente do site, o objetivo inicial tornou-se mais ambicioso e passou a ser a divulgação das Web Standards.

No início de 2006, criou o Blog do Maujor, hospedado em *http://www.maujor.com/blog/*, com propósito semelhante ao do site, mas com uma dinâmica mais ativa e a efetiva participação de seus leitores.

Maujor, como é conhecido o autor, é ainda um ativo frequentador de fóruns, escreve para vários portais brasileiros voltados a desenvolvedores web, é autor de artigos em inglês e de trabalhos relacionados às CSS publicados em sites internacionais sobre Web Standards.

Introdução

Este livro tem por objetivo fornecer aos profissionais envolvidos com o desenvolvimento para a web os conceitos fundamentais e as técnicas de programação necessárias ao desenvolvimento de interfaces responsivas com uso do framework Bootstrap.

Para quem foi escrito este livro

Este livro foi escrito para aquelas pessoas envolvidas na criação de sites tanto na área de design quanto na de desenvolvimento e programação, que desejam usar um framework mundialmente consagrado para criar interfaces usáveis e acessíveis em qualquer dispositivo, independentemente de suas características, isto é, que se adaptem às mais variadas resoluções de tela.

O objetivo do livro é fornecer informações detalhadas dos componentes do framework Bootstrap, estudando seus princípios e detalhando as funcionalidades previstas não somente pelo core (coração) do framework como também por cada um dos seus componentes. Explicações teóricas em linguagem corrente e clara, dispensando, sempre que possível, o jargão técnico avançado são acompanhadas de exemplos práticos explicados passo a passo e complementados por arquivo HTML para consulta.

Para tirar o máximo proveito dos ensinamentos contidos em cada capítulo é pré-requisito um razoável conhecimento de marcação HTML e das CSS e da criação de folhas de estilo. É desejável que o leitor tenha conhecimentos necessários para criar, com uso daquelas duas tecnologias, um layout simples de duas colunas.

Os conceitos e o entendimento das técnicas de desenvolvimento de interfaces responsivas são poderosas ferramentas de apoio na criação de sites mais acessíveis, enriquecendo a experiência do usuário. Profissionais da área de design, familiarizados com as técnicas aqui descritas, contarão com uma valiosa fonte de conhecimento e conceitos a empregar no planejamento das funcionalidades para incrementar suas criações, tornando-as universais.

Os iniciantes irão se beneficiar deste livro por começar seus estudos em uma fonte que aborda os princípios básicos de uso de frameworks CSS e das mais modernas técnicas de escrita do código, ensejando uma mudança no rumo de seu estudo que irá reduzir a curva de aprendizado e acelerar tal processo. Não se intimidem com conceitos ou terminologias que lhes sejam completamente desconhecidos nos primeiros capítulos. Com a sequência da leitura, as dúvidas tenderão a desaparecer naturalmente.

Convenções tipográficas

Com a finalidade de destacar diferentes tipos de informação neste livro, adotaram-se algumas convenções tipográficas mostradas a seguir.

Dica

Texto contendo uma dica sobre o assunto tratado:

Dica: Para suporte nativo das funcionalidades mostradas neste capítulo, consulte o site *http://caniuse.com*.

Alerta

Texto contendo um lembrete sobre procedimento extra com relação ao assunto tratado:

Alerta: Convém ressaltar que o grid do Bootstrap foi criado segundo os princípios da filosofia mobile first e, logicamente, é responsivo.

Terminologia

Texto estabelecendo a adoção de grafia-padrão em todo o livro para termos ou frases com mais de uma terminologia, uma tradução ou um significado:

Terminologia: Nos códigos desenvolvidos neste livro, adotou-se a sintaxe-padrão.

Chamada

Uma chamada para uma seção anterior na qual o assunto em questão foi explicado com detalhes.

Por exemplo:

Conforme estudamos no item [3.5.1.2], por padrão, os botões são estilizados com um padding que determina suas dimensões.

Neste exemplo, a chamada é para a seção 3.5.1.2, ou seja, capítulo 3 (o primeiro número indica o capítulo), item 5, subitem 1 subitem 2.

Marcação e scripts

Nas marcações e nos scripts que exemplificam a teoria, transcreveram-se somente os trechos que interessam ao assunto tratado. Omitiram-se os trechos que não dizem respeito ou não são relevantes para o entendimento do assunto, para não ocupar espaço desnecessário no livro.

Blocos de marcação são marcados com fonte monoespaçada:

```
<ul>
   <li><a href="#">Home</a></li>
   <li><a href="#">Portfólio</a></li>
   <li><a href="#">Loja</a></li>
   <li><a href="#">Contato</a></li>
</ul>
```

Trechos de marcação que merecem destaque são marcados em negrito:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../bootstrap/js/jquery.min.js">
   <\/script>')</script>
<script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
   <script type="text/javascript">
      $('.dropdown-toggle').dropdown();
   </script>
<body>
<html>
```

Para explicar passo a passo, quando julgado conveniente, cada linha do script foi apresentada com suas linhas numeradas e, a seguir, os comentários foram referenciados ao número da linha comentada:

- **HTML**

```
1.  <button type="button" class="btn btn-primary btn-lg" data-toggle="modal" data-
    target="#modalLogin">
2.     Abrir janela modal
3.  </button
4.  <div  class="modal fade" id="modalLogin" tabindex="-1" role="dialog" aria-
    labelledby="modalLoginLabel">
5.     <div class="modal-dialog" role="document">
6.         <div class="modal-content">
7.            <div class="modal-header">>
8.                <button type="button" class="close" data-dismiss="modal"
                      aria-label="Close"> <span aria-hidden="true">&times;</span></button>
9.                <h4 class="modal-title">Área administrativa</h4>
10.               </div>
11.               <div class="modal-body">
12                <!-- conteúdo da janela modal -->
13.             </div>
14.               <div class="modal-footer">
15.                   <button type="button" class="btn btn-default" data-dismiss="modal">
                          Fechar</button>
16.            </div>
17.        </div> <!-- /.modal-content -->
18.        </div> <!-- /.modal-dilog -->
19. </div> <!-- /.modal -->
```

[.../c5/modal-data-toggle.html]

Código comentado:

Linha(s)	Descrição
Linhas 1 a 3	Botão para abertura da janela modal. O atributo `data-target` deve apontar para o valor do atributo `id` do container geral da janela; em nosso caso, o `div#modalLogin` na linha 4. Opcionalmente, em lugar de um botão para abrir a janela, poderíamos usar um link, e neste caso o atributo `href` do link apontaria para o `id` do container geral da janela; em nosso caso, `href="#modalLogin"`. A classe `btn` definida para o link de abertura da janela faz com que os mecanismos internos do Bootstrap transformem o link em um botão. Esse botão pode ser estilizado com uso de classes adicionais, tal como `btn-primary` e `btn-lg`. A classe `data-toggle`, definida para aquele link, informa ao Bootstrap que a janela modal terá seu script de abertura e fechamento inserido automaticamente sem necessidade de codificação JavaScript pelo autor.

Linha(s)	Descrição (cont.)
Linha 4 a 19	Marcação do `div` container da janela modal. A classe `modal` é de uso obrigatório e informa ao Bootstrap que trata-se do container de uma janela modal. A classe `fade` faz com que o efeito de abertura seja por deslizamento da janela no sentido vertical, na direção de cima para baixo e com efeito de esmaecimento. Opcionalmente, podemos omitir o valor `fade` que produz o efeito de mostrar e esconder a janela sem deslizamento e esmaecimento, ou seja, abertura e fechamento fixa na sua posição. Sugiro fazer uma cópia do exemplo e retirar a classe `fade` verificando na prática os dois efeitos de abertura/fechamento. O atributo `id` é de livre escolha do autor.
Linhas 5 e 6	Containers auxiliares para estilização e scripts.

Arquivos para download

Os scripts mostrados no livro estão disponíveis para consulta online em *http://livrosdomaujor.com.br/bootstrap3/codigos.html*. Os arquivos foram nomeados com sintaxe própria, conforme o exemplo a seguir:

http://livrosdomaujor.com.br/bootstrap3/codigos/c4/botoes-dropdown.html

Este é o URL (endereço na web) do arquivo que mostra um exemplo contido no capítulo 4 do livro e referente à funcionalidade dropdown para botões.

Adicionalmente, ao final da marcação, há uma indicação do endereço para o arquivo conforme convenção mostrada no exemplo a seguir:

- **HTML**

```
<div class="btn-group">
   <button type="button" class="btn btn-default>EDITAR</button>
   <button type="button" class="btn btn-default dropdown-toggle"
      data-toggle="dropdown" aria-expanded="false">
      EDITAR <span class="caret"></span>
   </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">novo</a></li>
         <li><a href="#">abrir</a></li>
         <li><a href="#">fechar</a></li>
         <li class="divider"></li>
         <li><a href="#">salvar</a></li>
      </ul>
</div>
```

[.../c4/botoes-dropdown.html]

Destaques em geral

Palavras ou termos cujo significado deva ser destacado são grafados em itálico.

Por exemplo:

O Bootstrap oferece ainda alguns *grupos de classes* para estilização avançada de tooltip em forma diversa daquela do tooltip básico.

Variáveis

Valores variáveis em códigos são grafados em itálico.

Arquivos dos exemplos do livro

Os links para os arquivos dos exemplos deste livro constam de uma página web localizada em *http://www.livrosdomaujor.com.br/bootstrap3/codigos.html*.

Por tratar-se de um livro oferecido gratuitamente, tanto o autor como a editora se desobrigam de prestação de qualquer tipo de suporte técnico aos leitores.

CAPÍTULO 1

Apresentação do Bootstrap

Neste capítulo, faremos a apresentação do framework Bootstrap. Veremos a definição, o histórico e as razões para o uso do Bootstrap. Listaremos as seções e seus conteúdos onde constam os assuntos a serem tratados no livro. Mostraremos como criar o template mínimo para desenvolver uma aplicação, apresentaremos o Bootlint e abordaremos o suporte e a acessibilidade do framework.

Ao longo dos textos e das explicações contidas no livro, para não repetir exaustivamente o termo "framework Bootstrap", o abreviaremos para simplesmente *Bootstrap*. Assim, neste livro, salvo indicação contrária, *Bootstrap* é o mesmo que *framework Bootstrap*.

1.1 Introdução

Bootstrap é o mais popular framework JavaScript, HTML e CSS para desenvolvimento de sites e aplicações web responsivas e alinhadas com a filosofia mobile first. Torna o desenvolvimento front-end muito mais rápido e fácil. Indicado para desenvolvedores de todos os níveis de conhecimento, dispositivos de todos os tipos e projetos de todos os tamanhos.

1.1.1 Definição

O site do Bootstrap o define como um poderoso, elegante e intuitivo framework front-end que possibilita um desenvolvimento web de modo ágil e fácil.

1.1.2 Histórico

No dia 19 de agosto de 2011, Mark Otto (@*mdo*), um desenvolvedor trabalhando no Twitter, criador do Bootstrap juntamente com Jacob Thornton (@*fat*), anunciou ao mundo o lançamento do Bootstrap em um artigo publicado no blog do Twitter.

Iniciou o artigo dizendo estar feliz com o lançamento de um kit de ferramentas front-end destinado a agilizar o desenvolvimento de aplicações web.

Naquele artigo, relatou que no início do Twitter cada desenvolvedor usava a biblioteca que lhe era familiar para solucionar os problemas de front-end. Esse procedimento criou inconsistências, dificultando a integração, a escalabilidade e a manutenção das aplicações criadas por diferentes desenvolvedores da equipe.

Bootstrap foi a proposta de solução para aquelas inconsistências. Apresentada na primeira Twitter Hackweek realizada na semana de 22 a 29 de outubro de 2011, de lá saiu com sua primeira versão estável e pronta para uso.

1.1.3 Por que usar Bootstrap

No artigo de lançamento do Bootstrap, Mark Otto, ao explicar as razões para usar Bootstrap, começa afirmando que o coração do Bootstrap nada mais é do que CSS criado com LESS, um pré-processador destinado a gerar folhas de estilos CSS capaz de oferecer muito mais flexibilidade e poder do que as folhas de estilos convencionais, ou não processadas. Posteriormente foram criadas funcionalidades que permitem usar não somente LESS, mas também SASS como pré-processador CSS para o Bootstrap.

LESS e SASS são capazes de oferecer uma vasta gama de funcionalidades, tais como declarações CSS aninhadas, variáveis para valores de propriedades CSS, mixins (espécie de classe capaz de ser reusada), operadores e funções para declaração de cores.

Uma vez compilado via LESS ou SASS, o resultado é uma folha de estilos pura, cuja implementação na aplicação é muito simples; basta lincar o arquivo CSS ao documento. Outro benefício é que não há imagens, Flash ou JavaScript adicionais, somente folha de estilos.

Não é do escopo deste livro detalhar o uso dos pré-processadores LESS e SASS para criar folhas de estilos para o Bootstrap.

1.1.4 Seções da documentação do Bootstrap

Para fins de estudo, adotaremos neste livro as mesmas cinco seções constantes do site do Bootstrap, conforme listadas a seguir.

- **Introdução** – Seção de apresentação dos fundamentos ou da fundação inicial para a criação com uso do Bootstrap. Nessa seção são listadas as ferramentas de desenvolvimento, os arquivos do download do Bootstrap e a criação do ambiente de desenvolvimento. São mostrados os templates HTML5 para criação de aplicações em layouts fixos, fluidos e responsivos.
- **CSS** – Nessa seção são mostradas as regras CSS comuns, tais como definições de valores CSS para tipografia em geral, CSS reset etc. São ainda detalhadas as declarações CSS para vários elementos da marcação HTML, tais como tabelas, formulários, botões, imagens, uso de ícones e muito mais.
- **Componentes** – Nessa seção se estuda a criação de componentes de interface. Os componentes de interface previstos no Bootstrap são listados a seguir.
  - Glifos
  - Dropdown
  - Agrupamento de botões
  - Dropdown com botões
  - Grupamento de inputs
  - Navegação
  - Barra de navegação
  - Breadcrumbs (caminho da navegação)
  - Paginação
  - Rótulos e marcadores
  - Badges (marca de destaque)
  - Jumbotron (tela de destaque)
  - Page header (elemento `h1`)
  - Thumbnails
  - Alertas
  - Barra de progresso

- Objetos de mídia
    - Grupamento de listas
    - Painéis
    - Embed responsivo
    - Wells (container estilizado com bordas arredondadas, cor de fundo e efeito inset)
- **Plugins** – Nessa seção se estuda a integração dos plugins JavaScript com o Bootstrap. Os componentes gerados pelos plugins e as considerações sobre o uso da linguagem previstos no Bootstrap são listados a seguir.
    - Visão geral
    - Transições
    - Janelas modais
    - Dropdown
    - Scrollspy
    - Abas
    - Tooltip
    - Popover
    - Alertas
    - Botões
    - Collapse
    - Carousel
    - Affix
- **Personalização** – Trata-se de uma ferramenta online para fazer download personalizado de acordo com as necessidades específicas de um projeto. Essa ferramenta encontra-se em *http://getbootstrap.com/customize/* e seu uso é bem fácil e intuitivo.

Cada uma das seções listadas anteriormente será objeto de estudo e detalhamento neste livro.

1.2 Ferramentas de desenvolvimento

Para desenvolver um layout e criar os componentes gráficos de uma aplicação web, você precisa de um editor de texto, um navegador web e obviamente a estrutura de arquivos do Bootstrap. Se você optou por estudar o assunto proposto neste livro, supõe-se que tem um conhecimento razoável de marcação HTML e folhas de estilos CSS, assuntos estes que não fazem parte do escopo do livro.

Não vou indicar um editor de texto e navegador web, pois como você conhece HTML e CSS, já usa um deles. Assim, use o editor de texto e o navegador de sua preferência.

1.3 Download

Na homepage do site do Bootstrap, em *http://getbootstrap.com*, você encontra um link para download do Bootstrap.

São disponibilizadas três versões para download denominadas *Bootstrap*, *Source* e *SASS*. A versão *Bootstrap* é a mínima requerida para criação com uso de todas as funcionalidades do Bootstrap, e é ela que usaremos neste livro. A versão *Source* inclui toda a documentação do Bootstrap, e a versão *SASS* é para uso com o pré-processador.

Faça o download da versão *Bootstrap*, descomprima o arquivo compactado em uma pasta de sua livre escolha, e você terá a estrutura de pastas e arquivos, como mostrado na figura 1.1.

Na figura 1.1, repare que existem arquivos que têm uma forma compactada (minificada – *.min*) e uma forma não compactada. A forma não compactada é para uso opcional em fase de produção, pois são arquivos comentados e estruturados visualmente para fácil leitura por humanos.

Depois de lançado o projeto, esses arquivos deverão ser substituídos pelos seus equivalentes compactados, que são mais leves e de carregamento mais rápido, mas não se esqueça de compactar o arquivo caso você tenha feito alguma modificação no original, o que é pouco provável.

A pasta-raiz obtida com a descompactação do arquivo zipado do download denomina-se *bootstrap-versão-dist*, em que *versão* é a versão atual do Bootstrap (por exemplo, *bootstrap-3.3.5-dist* para o download da versão 3.3.5). Na pasta-raiz encontram-se três subpastas – *css*, *fonts*, *js* –, e dentro delas, os arquivos, conforme mostrados na figura 1.1.
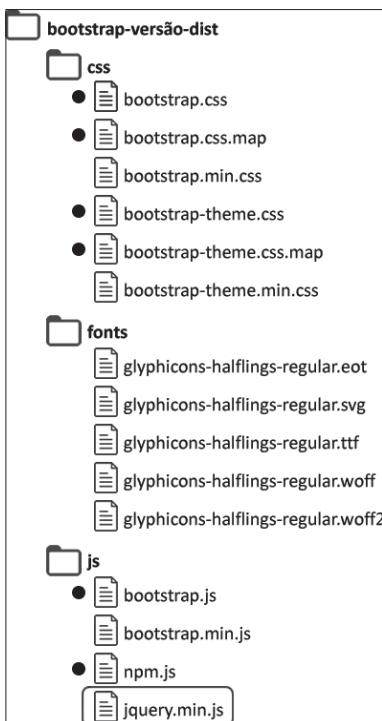
Figura 1.1 – Arquivos do download do Bootstrap.

Observe na figura que alguns arquivos foram destacados com a marca de uma bolinha. Tais arquivos podem ser apagados da pasta caso você não os use em fase de produção. Se usá-los, apague depois de lançado o projeto, pois, como dito anteriormente, devem ser substituídos pelos seus correspondentes minificados.

A seguir alguns comentários sobre a finalidade dos arquivos.

- *bootstrap.css* é o arquivo principal de estilização do Bootstrap em versão não comprimida. Usa-se a mesma versão comprimida (ou minificada) *bootstrap.min.css*. Se você pretende desenvolver uma folha de estilos personalizada, dê a ela um nome tal como *custom.css*, e quando terminar, comprima o arquivo. Atrele sua folha de estilo personalizada ao documento criando um link para ela depois do link que aponta para *bootstrap.min.css*.
- *bootstrap-theme.css* é um arquivo de estilização do Bootstrap que adiciona efeitos especiais de sombras, gradientes, 3D etc. em componentes tais como botões, painéis, caixas de destaque etc. Usa-se a mesma versão comprimida (ou minificada) *bootstrap-theme.min.css*. Se você pretende adicionar tais efeitos especiais em sua interface, use este arquivo. O arquivo *bootstrap-theme.min.css* é um arquivo de uso facultativo. Nos exemplos constantes deste livro, usaremos

esta folha de estilos. Para verificar seus efeitos na prática, comente na marcação da seção `head` do exemplo, o link que aponta para essa folha de estilos.

Alerta: Caso sua opção seja por usar os efeitos especiais de estilização, o link para o arquivo *bootstrap-theme.min.css* deverá vir na marcação HTML depois do link para o arquivo *bootstrap.min.css*.

- *bootstrap.css.map* e *bootstrap-theme.css.map* são arquivos destinados a mapear arquivos CSS minificados e fazê-los legíveis para humanos em ferramentas de debug, tipo as ferramentas do desenvolvedor nativas dos navegadores. Usam-se somente em fase de desenvolvimento de funcionalidades e temas CSS e não têm efeito sobre o produto final (o site ou a aplicação) criado.
- *bootstrap.js* é o arquivo, em versão não comprimida, que contém os scripts que fazem funcionar os plugins nativos do Bootstrap. Usa-se a mesma versão comprimida (ou minificada) *bootstrap.min.js*. Se você pretende desenvolver scripts personalizados, dê ao arquivo um nome tal como *custom.js* e, quando terminar, comprima o arquivo. Atrele seus scripts personalizados ao documento criando um link para o arquivo depois do link que aponta para *bootstrap.js*.
- *npm.js* é o arquivo para instalação do Bootstrap com NPM (fora do escopo deste livro).

O arquivo denominado *jquery.min.js* em destaque na figura *não faz parte do download*. Caso seu projeto preveja uso de componentes que usam JavaScript (plugins nativos do Bootstrap), será necessário carregar a biblioteca jQuery *na versão 1.11.2 ou posterior*. Consulte o site da jQuery para decidir qual é a versão mais atual (e posterior à jQuery 1.11.2) indicada para os navegadores aos quais você quer servir seu projeto.

Assim, deve-se, obrigatoriamente, fazer o download da biblioteca no site *http://jquery.com* e ter o arquivo gravado na pasta *js* do download. Ele servirá como fallback para o caso de falha no link para um CDN, como veremos no item [1.4] adiante.

Essa é a configuração básica do Bootstrap necessária à criação de layouts fixos, fluidos e responsivos. Com apenas o arquivo CSS *bootstrap.min.css* lincado ao documento você poderá começar a explorar essa poderosa ferramenta, e com o progresso no estudo e a consequente necessidade de uso de componentes que dependem de JavaScript, irá precisar lincar também o arquivo *bootstrap.min.js*.

1.4 Templates HTML

O Bootstrap foi desenvolvido e otimizado para ser usado com marcação HTML5. Assim o template mínimo para criar uma aplicação que use o Bootstrap é mostrado a seguir.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <!-- Estas 3 meta tags devem obrigatoriamente ser as primeiras na seção head -->
      <!-- As demais devem vir depois delas -->
   <title>Template mínimo para uso do Bootstrap</title>
   <link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet"> <!-- CSS do Bootstrap -->
   <!-- HTML5 shim and Respond.js para suporte dos elementos HTML5 e das media queries ao IE8 -->
   <!-- Atenção: Respond.js não funciona em páginas carregadas com uso do protocolo file:// -->
   <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
   <![endif]-->
</head>
<body>
   <h1>Título da aplicação</h1>
</body>
</html>
```

Com este template é possível desenvolver uma aplicação web que use as funcionalidades do Bootstrap que dependem apenas de estilização, por exemplo: tipografia, botões, painéis, seção em destaque etc. Não é possível usar as funcionalidades que dependem dos plugins JavaScript nativos do Bootstrap, por exemplo: barras de navegação, janelas modais, slideshow etc.

Alerta: Nos exemplos constantes deste livro não será considerado o suporte para o IE8.

Caso você queira usar, no seu projeto, os 12 plugins nativos do Bootstrap, listados na seção JavaScript do item [1.1.4], deverá incluir no template, mostrado anteriormente, a biblioteca jQuery e o aquivo *bootstrap.min.js*, como mostrado em destaque a seguir.

```
...
<body>
   <h1>Título da aplicação</h1>
   <!-- jQuery (caso se use plugins JavaScript do Bootstrap) -->
   <script src="http://code.jquery.com/jquery-1.11.2.min.js" type="text/javascript">
   </script>
   <script type="text/javascript"> // fallback para jQuery
   //<![CDATA[
   (window.jQuery)||document.write('<script type="text/javascript"
      src="../bootstrap/js/jquery.min.js"><\/script>');//]]>
   </script>
   <!-- Bootstrap -->
   <script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

Conforme mostrado no código, é de boa prática usar um CDN para lincar para a biblioteca jQuery. Em nosso caso, escolhemos o CDN da jQuery. Caso o link para o CDN apresente qualquer problema, existe um link alternativo (fallback) para um arquivo da biblioteca jQuery hospedado no servidor da aplicação e colocado na pasta *js* do download, conforme mostrado e comentado na figura 1.1.

Alternativamente, você pode usar o CDN Google, apontando para:

http://ajax.googleapis.com/ajax/libs/jquery/ultima-versao/jquery.min.js

ou o CDN da Microsoft, apontando para:

http://ajax.aspnetcdn.com/ajax/jquery/jquery-ultima-versao.js

---

Alerta: Ao optar por um CDN, confira se a versão que você quer usar já está hospedada no CDN. Cole o endereço do arquivo na caixa de endereços do navegador e certifique-se de que o link não está "quebrado" e aponta para o arquivo pretendido.

Opcionalmente, você poderá usar o CDN do Bootstrap (*http://www.bootstrapcdn.com/*) para lincar, também, os arquivos do Bootstrap como mostrado a seguir:

```
<!-- Versão compilada e minificada das CSS -->
<link rel="stylesheet"
   href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<!-- Tema opcional -->
```

```
<link rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">

<!-- Versão compilada e minificada da JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js">
</script>
```

Convém notar que o arquivo *bootstrap.min.js* contém *todos* os plugins do Bootstrap. É muito provável que o seu projeto use somente alguns deles. Para estas situações existe a possibilidade de personalizar o download, obtendo somente os plugins necessários ao projeto.

Convém notar que nestas situações devemos usar a versão do download personalizado e não um CDN que fornece todos os plugins.

Para personalizar o download dos plugins e também das CSS e dos componentes do Bootstrap, visite a página *http://getbootstrap.com/customize/* no site do Bootstrap.

1.5 Bootlint

Bootlint é uma ferramenta de desenvolvimento, oficial do Bootstrap, destinada a examinar a marcação HTML criada para uma aplicação ou um site desenvolvido com uso de Bootstrap e informar aos desenvolvedores possíveis “bugs” ou não conformidades encontradas na marcação.

A ferramenta é fornecida em forma de bookmarklet (para ser instalada em um navegador); seu código e as instruções de instalação podem ser encontrados no GitHub em *http://kwz.me/I4*.

Das diversas formas de instalação, a mais simples e imediata é arrastar o código JavaScript do bookmarklet, que se encontra no GitHub sob o título “Getting Started”, subtítulo “In the browser”, para a barra de favoritos do navegador.

Depois de instalá-la como favorito no navegador, abra a página web que contém a marcação HTML para o Bootstrap e abra também o console do navegador. Limpe o console e clique no bookmarklet criado nos favoritos. Como resultado será aberta uma janela tipo alerta da JavaScript informando sobre a existência ou não de “bugs” na página.

Feche a janela e, caso existam “bugs”, eles serão descritos sumariamente no console do navegador e um código será fornecido. No GitHub, em *http://kwz.me/I0*, você encontra a descrição de cada código.

É altamente recomendável instalar e usar esta ferramenta no seu ambiente de desenvolvimento verificando constantemente as marcações criadas e não deixando para verificar tudo no final do projeto.

1.6 Modelo CSS e o box-sizing

O Bootstrap declara, em sua folha de estilos, o moderno Modelo CSS modificado usando a propriedade CSS `box-sizing` de modo a excluir `padding` e `margin` das dimensões dos boxes, conforme a regra de estilos mostrada a seguir.

```
* {
   -webkit-box-sizing: border-box;
      -moz-box-sizing: border-box;
         box-sizing: border-box;
}
*:before,
*:after {
   -webkit-box-sizing: border-box;
      -moz-box-sizing: border-box;
         box-sizing: border-box;
}
```

Esta declaração pode conflitar com softwares de terceiros, tais como Google Maps e Google Search Engine, inseridos em uma página que use o Bootstrap, pelo fato de tais softwares usarem em suas CSS o Modelo CSS tradicional.

Assim, esteja ciente de que se o layout em sua página "quebrar" por causa da inserção de um widget ou outro software qualquer pertencente a terceiros, será necessário que você localize e isole a parte da marcação HTML onde o problema ocorre e crie uma folha de estilo específica para reverter o modelo CSS naquele trecho da marcação, ou, em lugar de uma folha de estilos específica, adicione regras CSS, conforme sugeridas a seguir, na folha de estilos do Bootstrap.

Como regra geral, existem duas soluções para estes casos. A primeira, reverter o modelo CSS para um elemento da marcação; e a segunda, para uma determinada região da marcação, conforme sugerido a seguir.

```
/* Opção 1: para um elemento da marcação */
.element {
   -webkit-box-sizing: content-box;
      -moz-box-sizing: content-box;
```

```
        box-sizing: content-box;
}
/* Opção 2: para uma região da marcação cujo elemento ancestral deve receber a classe
reset-box-sizing */
.reset-box-sizing,
.reset-box-sizing *,
.reset-box-sizing *:before,
.reset-box-sizing *:after {
   -webkit-box-sizing: content-box;
      -moz-box-sizing: content-box;
         box-sizing: content-box;
}
```

1.7 Acessibilidade

O Bootstrap contém funcionalidades que facilitam o cumprimento das diretrizes de acessibilidade previstas pelo W3C.

Para criar um link que permita às tecnologias assistivas pular o menu de navegação e ir para o conteúdo, são previstas as classes `sr-only` e `sr-only-focusable`, que devem ser aplicadas a uma marcação-padrão, conforme mostrado a seguir.

```
<a href="#conteudo" class="sr-only sr-only-focusable">Pular para o conteúdo</a>
   <!-- menu de navegação -->
<div id="conteudo" tabindex="-1">
   <!-- Conteúdo principal da página -->
</div>
```

A marcação mostrada, quando for o caso de se usar, deve ser colocada no início da página. As classes `sr-only` e `sr-only-focusable` são nativas do Bootstrap.

A classe `sr-only` faz com que o texto do link não seja visível na tela, mas "visto" por tecnologias assistivas, tais como leitores de tela.

Para usuários sem restrições visuais que navegam com auxílio do teclado, a classe `sr-only-focusable` faz com que o texto do link se torne visível quando o link recebe o foco.

O par atributo/valor `tabindex="-1"` corrige bugs de foco em alguns navegadores.

O exemplo prático mostrado a seguir cria um link tipo "Pular navegação" e encontra-se disponível para consulta online. Não se preocupe com a marcação HTML que não esteja em destaque. Ela é própria do Bootstrap e será estudada no próximo capítulo.

```
<a href="#conteudo" class="sr-only sr-only-focusable">Pular para o conteúdo</a>
...
<div class="row">
   <div class="col-md-6">
   <h3>Menu de navegação</h3>
   <p>Extensa lista de links</p>
   <ul>
      <li><a href="#">link</a></li>
         <!-- mais links -->
   </ul>
</div> <!-- /..col-md-6 -->
</div> <!-- /.row -->

<div class="row">
<div class="col-md-6">
   <h1 id="conteudo" tabindex="-1">Conteúdo principal da página</h1>
      <p>bla...bla...bla...bla</p>
         <!-- mais conteúdo -->
</div> <!-- /.col-md-6-->
</div> <!-- /.row -->
```

[.../c1/pular-navegacao.html]

Abra este arquivo em um navegador e pressione a tecla Tab. Essa ação irá revelar no topo da página o texto do link. A seguir, tecle Enter para navegar diretamente para o conteúdo da página.

1.8 Suporte

O Bootstrap foi criado para funcionar em navegadores modernos tanto em desktop como em dispositivos móveis. Navegadores antigos falharão na estilização, mas a parte funcional da página será preservada.

Quanto ao Internet Explorer, o suporte é consistente nas versões IE9 e superiores. Se você pretende servir sua aplicação para o IE8, deve incluir os scripts *shim* e *Respond.js* que se destinam a possibilitar que aquele navegador reconheça os elementos da HTML5 e as regras media queries respectivamente, pois ele não oferece suporte nativo para tais funcionalidades. Não é do escopo deste livro detalhar as implicações e os detalhes de funcionamento e inclusão destes scripts.

Os exemplos constantes deste livro não consideram o suporte para o IE8.

Observe as tabelas 1.1 e 1.2 detalhando o suporte ao Bootstrap por dispositivos e navegadores.

Tabela 1.1 – Suporte ao Bootstrap por navegadores

	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	SIM	SIM		NÃO	N/A
iOS	SIM	N/A	N/A	NÃO	SIM
Mac OS X	SIM	SIM		SIM	SIM
Windows	SIM	SIM	SIM	SIM	NÃO

Tabela 1.2 – Suporte ao Bootstrap pelos IE8 e IE9

Funcionalidade	Internet Explorer 8	Internet Explorer 9
border-radius	NÃO	SIM
box-shadow	NÃO	SIM
transform	NÃO	SIM, com prefixo -ms
transition	NÃO	
placeholder	NÃO	

Para suporte às funcionalidades HTML5 e CSS3 em geral, consulte o site caniuse (*http://kwz.me/I5*).

Isso é tudo que você precisa para começar o estudo do Bootstrap. Nos próximos capítulos, detalharemos as funcionalidades do Bootstrap desenvolvendo exemplos práticos que mostram desde a criação de simples componentes de interfaces até o desenvolvimento de layouts complexos.

Para os exemplos constantes deste livro, criamos uma pasta-raiz denominada *codigos* e, dentro dela, uma subpasta denominada *bootstrap* contendo o diretório e os arquivos obtidos do Bootstrap, ou seja, renomeamos a pasta *bootstrap-versão-dist* mostrada na figura 1.1 para simplesmente *bootstrap* e mais subpastas para os capítulos (*c1*, *c2*, *c3*, *c4* etc.) contendo os arquivos dos exemplos, tudo conforme mostrado na figura 1.2.
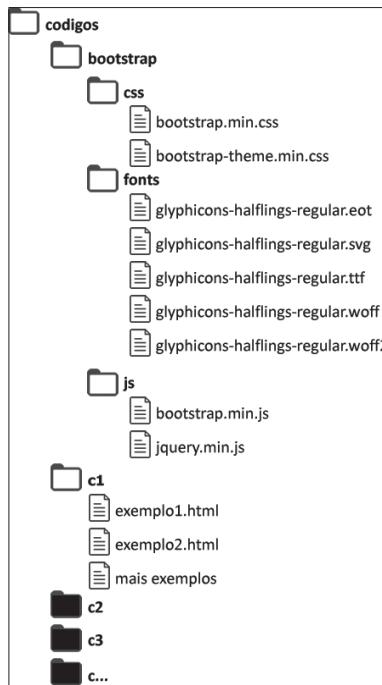
Figura 1.2 – Estrutura de pastas para arquivos dos exemplos do livro.

CAPÍTULO 2

Fundamentos

Neste capítulo estudaremos os fundamentos do Bootstrap. Será mostrada a normalização via reset de valores de propriedades CSS. Mostraremos a criação de um grid fixo para desenvolvimento de layouts e, no final do capítulo, estaremos cientes dos conceitos básicos de frameworks CSS em geral e em condições de começar o estudo da estrutura do grid e das funcionalidades do Bootstrap.

2.1 Marcação HTML

O Bootstrap foi criado para ser usado com a linguagem de marcação HTML5. As suas funcionalidades foram desenvolvidas considerando certos elementos exclusivos daquela linguagem, bem como de propriedades CSS avançadas que requerem o uso da HTML5.

Assim, a marcação HTML de uma aplicação deverá obrigatoriamente conter o DOCTYPE da HTML5 como mostrado a seguir.

```
<!DOCTYPE html>
<html lang="pt-br">
   ...
</html>
```

2.2 CSS reset

O termo reset no contexto de programação pode ser traduzido, usando linguagem não técnica, como a ação de "zerar" valores previamente definidos.

CSS reset é uma técnica inventada por Eric Meyer (*http://meyerweb.com/eric/tools/css/reset/*) no ano de 2007, amplamente usada desde então, que consiste em se definir uma folha de estilos cuja finalidade é "zerar" valores CSS padrão dos navegadores.

Todo navegador tem uma folha de estilos nativa que define as propriedades CSS iniciais dos elementos da marcação. Você poderá visualizar essa folha de estilos em ação ao criar um documento onde faça constar, por exemplo, cabeçalhos (`h1`, `h2`, `h3` etc.), parágrafos (`p`), listas (`ol`, `ul`, `dl`) etc. Ao visualizar o documento em um navegador, note que os conteúdos dos elementos da marcação têm uma estilização inicial, sem que você tenha declarado qualquer folha de estilos. A cor de fundo do elemento `body` é branca, os textos são em preto, cabeçalhos são em negrito com tamanhos de fonte diferentes, a fonte é da família `serif`, existem margens entre cabeçalhos e parágrafos, listas são endentadas, e por aí vai.

Se você decidiu criar o documento sugerido, experimente visualizá-lo em diferentes navegadores e compare as renderizações. Vai notar, nos diferentes navegadores, que muitos dos estilos aplicados são os mesmos, tais como cor de fundo branca do elemento `body`, preta dos textos, negrito em cabeçalhos, família `serif` para fontes etc. Notará também que a renderização não é exatamente a mesma, por exemplo, quando se trata do posicionamento e do tamanho dos textos na página. Isso acontece porque cada navegador adota sua folha de estilos própria e alguns valores CSS nessas folhas, tais como valores para as propriedades `margin`, `line-height`, `font-size` etc., diferem de um navegador para outro. Isso em determinados casos pode ser desastroso para o layout e pode exigir do desenvolvedor trabalho extra para tratar das inconsistências entre navegadores resultantes das diferenças entre folhas de estilos nativas.

A finalidade das CSS reset, que nada mais são do que um conjunto de declarações de estilos (uma folha de estilos), é padronizar, para os navegadores, a renderização inicial dos elementos da marcação, zerando os valores de algumas propriedades CSS, tais como `margin` e `padding`, e estabelecendo um valor-padrão para outras, tais como `line-height`, `vertical-align`, `font-size`, sobrescrevendo a folha de estilos nativa do navegador e resolvendo o problema da inconsistência de estilização inicial entre navegadores.

No início de 2012, Nicolas Gallagher, um desenvolvedor trabalhando no Twitter, e Jonathan Neal criaram o projeto denominado *normalize.css* (*http://necolas.github.com/normalize.css/*), cuja proposta era disponibilizar para a comunidade de desenvolvedores front-end uma folha de estilos reset destinada a padronizar a renderização nativa dos navegadores. Na página do projeto, os autores discorrem sobre as diferenças e vantagens de usar *normalize.css* em relação as CSS reset de Eric Meyer.

O Bootstrap, o projeto HTML5 Boilerplate, a NASA, entre outras grandes companhias, atualmente usam o *normalize.css*. Mesmo que você já conheça as

técnicas de CSS reset, aconselho a abrir a página contendo a folha de estilos de *normalize.css* clicando o link para download do arquivo no endereço web *http://necolas.github.com/normalize.css/*. Inspecione atentamente as declarações CSS, especule consigo mesmo sobre a finalidade de cada uma delas e acrescente mais esse conhecimento ao seu acervo profissional.

A folha de estilos *normalize.css* foi incorporada dentro da folha de estilo do Bootstrap. Assim, você não precisa criar link para o arquivo *normalize.css*, pois ele já consta da folha de estilo nativa do Bootstrap.

2.3 Sistema de grid para layout

Criar layout CSS baseado em um grid inicial é uma técnica que surgiu em 2007 e proporcionou o aparecimento de grids prontos, publicados por seus autores, para uso pelos desenvolvedores web. Tais grids constituem os chamados frameworks CSS para criação de layouts ou simplesmente grids CSS, e o Bootstrap se baseia em grid.

Um dos sistemas de grid CSS pioneiro denomina-se Blueprint (*http://www.blueprintcss.org/*) e é usado até hoje. Logo foram lançados outros sistemas, e atualmente existem dezenas deles, sendo que um dos mais conhecidos dos autores CSS é o sistema 960 Grid Sistem (*http://960.gs/*).

Usar um framework baseado em grid para desenvolver layouts CSS é um assunto polêmico e não é do escopo deste livro discutir a conveniência ou enumerar aspectos negativos e positivos do seu uso. No entendimento deste autor, dois aspectos são indiscutíveis quando se trata de criar um layout baseado em grid *versus* a codificação "a unha": agilização do processo de criação e possibilidade de usar as funcionalidades do grid de forma modular.

A agilização do processo traz como benefício imediato a redução do tempo de trabalho para atingir o resultado final, e nós, desenvolvedores, sabemos muito bem como os prazos são curtos. Um framework para criação de layout bem projetado é de natureza modular, ou seja, possibilita ao autor usar estritamente as funcionalidades específicas ao seu projeto, sem a necessidade de carregar arquivos desnecessários.

Dica: Uma das características do Bootstrap é a sua natureza modular, materializada na existência de uma interface gráfica, no site do projeto, destinada a fazer o download personalizado do Bootstrap (*http://getbootstrap.com/customize/*).

O sucesso de um projeto web começa com um planejamento bem feito no qual a escolha das ferramentas e o detalhamento das fases do desenvolvimento são de importância fundamental. Assim, usar um framework grid ou codificar "a unha" é uma decisão que cabe à equipe de desenvolvimento.

Sem dúvidas, o inventor ou os inventores dos sistemas de grid CSS, tal como é conhecido atualmente, tiveram um ideia brilhante e digna de admiração, sem contar a enorme contribuição que fizeram ao desenvolvimento web colocando nas mãos dos autores uma ferramenta extremamente poderosa, ágil e fácil de aprender, destinada à criação de layouts CSS. Atualmente, uma quantidade considerável de sites foi desenvolvida com uso de grid CSS.

O W3C lançou em abril de 2011 um Rascunho de Trabalho denominado Grid Layout, renomeado em 22 de março de 2012 como CSS Grid Layout, descrevendo funcionalidades para criação de layouts baseados em grids. Essa especificação está no seu início de desenvolvimento e demandará algum tempo até que possa ser usada em produção. O curioso é que o Internet Explorer 10 é, desde dezembro de 2012, o único navegador que implementou as funcionalidades previstas naquela especificação, devendo-se usar o prefixo proprietário `-ms-` para declarar as regras CSS nela previstas. Posteriormente a especificação alterou a sintaxe, mas o IE continua a dar suporte à sintaxe antiga.

A ideia que inspirou o sistema de grid CSS é muito simples. Uma vez conhecido o mecanismo de funcionamento, somos imediatamente levados a fazer a clássica pergunta a nós mesmos: "Como eu não pensei nisso?".

Grid significa malha e tecnicamente pode ser descrito como uma estrutura bidimensional formada por eixos (ou linhas retas) verticais e horizontais que se interceptam, criando uma estrutura de malha com células definidas pelas quatro interseções de eixos verticais com eixos horizontais.

A diagramação dos conteúdos e a consequente criação do layout se faz com uso dessa malha com suas células servindo de container para conteúdos a diagramar (ou "layautar"). Células podem ser combinadas na horizontal e/ou na vertical de modo a criar containers de tamanhos diferentes.

A técnica de uso de grid em tipografia foi introduzida no currículo acadêmico da formação de designers na metade dos anos 70 e começou a ser usada para diagramar conteúdos destinados à mídia impressa. Somente por volta de 2007 a técnica foi usada para criação de layouts CSS, e hoje em dia está amplamente difundida nesta área.

A figura 2.1 mostra dois exemplos de estruturas de grids.

Figura 2.1 – Grids.

2.4 Como é criado um grid CSS

A marcação HTML destinada a ser estilizada de forma a constituir um grid CSS é uma marcação modulada e padrão na qual os elementos que constituem a estrutura do layout são definidos com uso de um atributo cujo valor é previamente definido e serve como seletor CSS para definir as células do grid, tais como definir células unitárias ou combinar células horizontais de modo a formar containers para o layout.

2.4.1 Container geral

Vamos examinar um exemplo prático bem simples de criação de grid com a finalidade de entender o funcionamento básico das estruturas para grid. Observe a marcação HTML contendo uma folha de estilo incorporada mostrada a seguir.

Dica: Nos exemplos deste livro, quando for o caso, usaremos folha de estilo incorporada com a finalidade de facilitar a leitura e o entendimento dos códigos. Você com certeza conhece e sabe quando, como e por que usar folhas de estilos externas.

- **HTML**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
```

```
<meta charset="utf-8">
<title>Livro Bootstrap Maujor</title>
<style>
   body { margin: 0; font: 100% sans-serif; }
   h1 { font-size: 160%; }
   .container { width: 940px; margin: 0 auto; background: #b5b5b5; }
</style>
</head>
<body>
   <div class="container">
      <h1>Meu primeiro grid - container geral</h1>
   </div> <!-- /.container -->
</body>
</html>
```

[.../c2/grid-basico-1.html]

A marcação HTML mostrada, estilizada conforme a folha de estilo nela incorporada, cria um container geral para um grid CSS fixo com largura de 940px centrado na tela.

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.2.
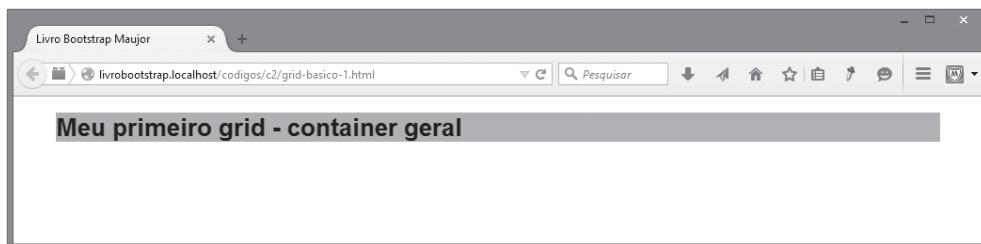
Figura 2.2 – Container para um grid básico.

2.4.2 Linha de células

Vamos criar uma linha de células para nosso grid. Para isso precisamos definir quantas células colocaremos em uma linha e qual será o espaçamento entre elas. Escolhemos colocar em cada linha doze células separadas por 20px. Fazendo uns cálculos simples, chegamos à conclusão de que cada célula deverá ter 60px de largura, totalizando 60 x 12 = 720px, com um total de espaços de 11 x 20 = 220px. Somando 720 + 220, obtemos os 940px do container do layout, "fechando" as contas.

Vamos transformar essas premissas em marcação HTML e CSS. A linha de células será constituída pelo elemento `div.row` e cada uma das células por um elemento `div.span1`. Observe a seguir o acréscimo de marcação no HTML mostrado anteriormente.

- **HTML**

```
1. <div class="container">
2.     <h1>Meu primeiro grid - linha de células</h1>
3.     <div class="row">
4.         <div class="span1">1</div>
5.         <div class="span1">2</div>
6.         <div class="span1">3</div>
7.         ...
8.         <div class="span1">11</div>
9.         <div class="span1">12</div>
10.    </div> <!-- /.row -->
11. </div> <!-- /.container -->
```

[.../c2/grid-basico-2.html]

Código comentado:

Linha(s)	Descrição
Linhas 3 a 10	Elemento container `div` com a classe cujo valor é `row` e que será o container das células. A tradução da palavra inglesa “row” é linha. Esse container é uma linha de células do grid.
Linhas 4 a 9	Doze elementos `div` com a classe cujo valor é `span1` são as doze células do grid.

Para dispor uma linha contendo doze células, acrescentaremos na folha de estilos mostrada anteriormente as declarações em destaque no código a seguir.

- **CSS**

```
1. <style tyep="text/css">
2. body { margin: 0; font: 100% sans-serif; }
3. h1 { font-size: 160%; }
4. .container {
5.     width: 940px;
6.     margin: 0 auto;
7.     background: #b5b5b5;
8. }
```

```
9.  .row { margin-left: -20px; }
10. .span1 {
11.     width: 60px;
12.     float: left;
13.     margin-left: 20px;
14.     background: #d5d5d5
15. }
16. .row:after {  /* técnica moderna para clear floats */
17.     content: "";
18.     display: table;
19.     line-height: 0;
20.     clear:both;

21. }
22. </style>
```

Código comentado:

Linha(s)	Descrição
Linha 9	Ver explicação na linha 13.
Linhas 11	Define a largura de cada célula.
Linhas 12	Flutua as células à esquerda, tirando-as da sua posição vertical padrão e colocando-as em linha ao lado da outra.
Linha 13	Define um espaçamento de 20px à esquerda para cada célula. Note que essa declaração fará com que a primeira célula da linha seja deslocada para a direita. Para recolocá-la em sua posição, declaramos uma margem negativa de 20px para o container da linha, como mostrado na linha 9.
Linha 14	A cor de fundo `#d5d5d5` visa apenas a facilitar a visualização do grid criando contraste com a cor do container geral `#b5b5b5` (ver linha 7).
Linhas 16 a 21	Uma vez que as células `div.span1` foram flutuadas dentro do seu container `div.row`, esse não se estende na vertical, assumindo altura zero. É o clássico comportamento de containers com floats. Para estender o container, precisamos declarar regras CSS para fazer o conhecido clear floats. Nesse exemplo, adotamos a moderna técnica clear floats sem uso de marcação.

A marcação HTML mostrada, estilizada conforme os acréscimos das regras CSS apresentadas, cria uma linha de doze células conforme a proposta de layout.

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.3.

Figura 2.3 – Grid básico, linha de células.

2.4.3 Unindo células

A próxima funcionalidade que o sistema de grid proporciona, e que estudaremos a seguir, é a possibilidade de unir duas ou mais células adjacentes para formar colunas. Por exemplo: unindo-se sucessivamente quatro células adjacentes, podemos criar uma linha composta de três colunas.

Alerta: A partir daqui, vamos renomear o que até aqui chamamos de célula. Em grids CSS, as chamadas células, bem como o resultado da união de células adjacentes, denominam-se colunas. Assim, no exemplo mostrado na figura 2.3, temos 12 colunas.

Vamos exemplificar a união de células do grid tomando como objetivo a criação de uma estrutura de layout de duas colunas.

Uma coluna esquerda para o conteúdo principal e uma coluna direita para o conteúdo auxiliar, tais como navegação, parceiros etc. É nossa intenção que a coluna principal ocupe 2/3 da largura total da aplicação (em nosso exemplo, 940px), e a coluna de navegação, 1/3 daquela largura.

Observe os acréscimos e as modificações na marcação HTML.

- **HTML**

```
1. <div class="container">
2.     <h1>Meu primeiro grid - criando colunas</h1>
3.     <div class="row">
4.         <div class="span8"><h2>Coluna principal</h2></div>
5.         <div class="span4"><h2>Coluna auxiliar</h2></div>
```

[.../c2/grid-basico-3.html]

Código comentado:

Linha(s)	Descrição
Linhas 3 e 10	Essas linhas definem o início e o final do container das células de uma linha do grid. Note que os elementos-filhos desse container são dois elementos `div` em lugar dos doze elementos `div.span1` que tínhamos no exemplo anterior.
Linhas 4 e 5	Os dois elementos `div`, na linha, criam as colunas principal e auxiliar. Note que o valor do atributo `class` desses `div` termina com um número (`span8` e `span4`). Esses dois números somados resultam em 12. Como você já deve ter concluído, o número no valor do atributo define o número de células a serem combinadas para formar a coluna.

O termo `span` em inglês, no presente contexto, pode ser traduzido para *unir*, *abarcar* ou *abranger*. Assim `div.span8` deve ser lido como: um elemento `div` que une oito células do grid para criar o container de uma coluna do layout.

Para criar as duas colunas do layout, modificaremos e acrescentaremos na folha de estilos mostrada anteriormente as declarações em destaque no código a seguir.

- **CSS**

```
1.  <style type="text/css">
2.  body { margin: 0; font: 100% sans-serif; }
3.  h1 { font-size: 160%; }
4.  .container {
5.      width: 940px;
6.      margin: 0 auto;
7.      background: #b5b5b5;
8.  }
9.  .row { margin-left: -20px; }
10. .span1 {
11.     width: 60px;
12.     float: left;
13.     margin-left: 20px;
14.     background: #d5d5d5
15. }
16. .row:after {
17.     content: "";
18.     display: table;
19.     line-height: 0;
20.     clear:both;
21. }
```

```
22. [class*="span"] {
23.     float: left;
24.     margin-left: 20px;
25.     background: #d5d5d5
26. }
27. .span4 { width: 300px; }
28. .span8 { width: 620px; }
29. </style>
```

Código comentado:

Linha(s)	Descrição
Linhas 10 a 15	A declaração da largura para cada seletor `span*` deverá ser feita em regra CSS separada, assim a regra CSS contida nessa linha que declara valores para o seletor `span1` será substituída por outras regras mostradas a seguir.
Linhas 22 a 26	Os valores CSS comuns a todos os seletores do tipo `span*` são aqui declarados. O seletor `[class*="span"]` aplica-se a todos os elementos cujo valor do atributo `class` seja uma palavra que começa com `span` (vale para `span1`, `span2`, `span3`, `span4` etc.).
Linha 27	Aqui se define a largura da coluna formada pela união de quatro células. A largura é obtida somando-se as larguras das células (60px) com a largura dos espaçamentos entre elas (20px), ou seja, 4 x 60px + 3 x 20px = 300px.
Linha 28	Aqui se define a largura da coluna formada pela união de oito células. A largura é obtida somando-se as larguras das células (60px) com a largura dos espaçamentos entre elas (20px), ou seja, 8 x 60px + 7 x 20px = 620px.

A marcação HTML mostrada, estilizada conforme os acréscimos e as modificações nas regras CSS mostradas, cria as duas colunas conforme proposto anteriormente.

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.4.
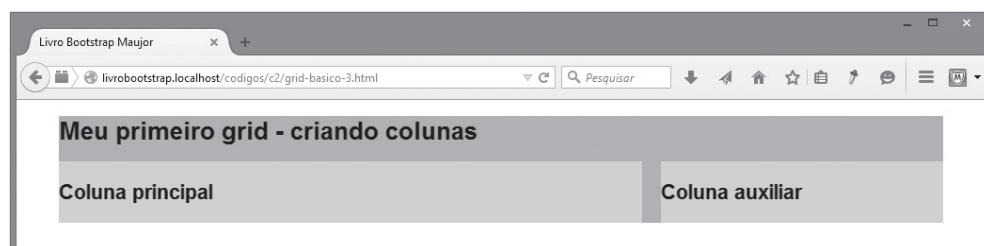
Figura 2.4 – Grid básico, criando colunas.

Para contemplar todas as possibilidades de combinação de união de células, a folha de estilos do grid deverá conter as seguintes declarações CSS:

- **CSS**

```
.span1  { width: 60px; }
.span2  { width: 140px; } /* 2x60 + 1x20 */
.span3  { width: 220px; } /* 3x60 + 2x20 */
.span4  { width: 300px; } /* 4x60 + 3x20 */
.span5  { width: 380px; } /* 5x60 + 4x20 */
.span6  { width: 460px; } /* 6x60 + 5x20 */
.span7  { width: 540px; } /* 7x60 + 6x20 */
.span8  { width: 620px; } /* 8x60 + 7x20 */
.span9  { width: 700px; } /* 9x60 + 8x20 */
.span10 { width: 780px; } /* 10x60 + 9x20 */
.span11 { width: 860px; } /* 11x60 + 10x20 */
.span12 { width: 940px; } /* 12x60 + 11x20 */
```

2.4.4 Criando um rodapé

Para criar um rodapé em nosso sistema de grid básico, basta declarar na marcação HTML uma nova linha `div.row` que se estenda por toda a largura do container do grid `div.container`. O acréscimo de marcação HTML e de regra CSS é mostrado, em destaque, a seguir.

- **HTML**

```
1.  <div class="container">
2.      <h1>Meu primeiro grid - criando rodapé</h1>
3.      <div class="row">
4.          <div class="span8"><h2>Coluna principal</h2></div>
5.          <div class="span4"><h2>Coluna auxiliar</h2></div>
10.     </div> <!-- /.row  das colunas -->
11.     <div class="row">
12.         <div class="span12"><h2>Rodapé</h2></div>
13.     </div> <!-- /.row  do rodapé -->
14. </div> <!-- /.container -->
```

[.../c2/grid-basico-4.html]

- **CSS**

```
.span12 { width: 940px; background: #e5e5e5; } /* Acréscimo na folha de estilo anterior
*/
```

Alerta: No arquivo para esse exemplo define-se uma cor de fundo para a linha do rodapé com a finalidade de destacar visualmente sua posição.

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.5.

Figura 2.5 – Grid básico, criando rodapé.

2.4.5 Aninhando conteúdos

Outra funcionalidade do sistema de grids para layout é a possibilidade de aninhar colunas dentro de colunas. Suponha que desejamos criar dentro da coluna principal três blocos de conteúdos em linha, ou seja, na horizontal. O acréscimo de marcação HTML e de regra CSS é mostrado a seguir.

- **HTML**

```
1.  <div class="container">
2.      <h1>Meu primeiro grid - aninhamento de colunas</h1>
3.      <div class="row">
4.          <div class="span8">
                <h2>Coluna principal</h2>
                    <div class="row">
                        <div class="span2" style="background: #f5f5f5;">
                            <p>Conteúdo aninhado á esquerda</p></div>
                        <div class="span3" style="background: #f5f5f5;">
                            <p>Conteúdo aninhado no centro</p></div>
                        <div class="span3" style="background: #f5f5f5;">
```

```
                    <p>Conteúdo aninhado à direita</p></div>
                 </div> <!-- .row dos conteúdos aninhados-->
         </div> <!-- /.span8 coluna principal -->
5.       <div class="span4"><h2>Coluna auxiliar</h2></div>
```

[.../c2/grid-basico-5.html]

- **CSS**

```
.span2 { width: 140px; } /* Acréscimo na folha de estilo anterior */
.span3 { width: 220px; } /* Acréscimo na folha de estilo anterior */
```

Alerta: No arquivo para esse exemplo define-se uma cor de fundo, com CSS inline, para os conteúdos aninhados, somente com a finalidade de destacar visualmente sua posição.

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.6.
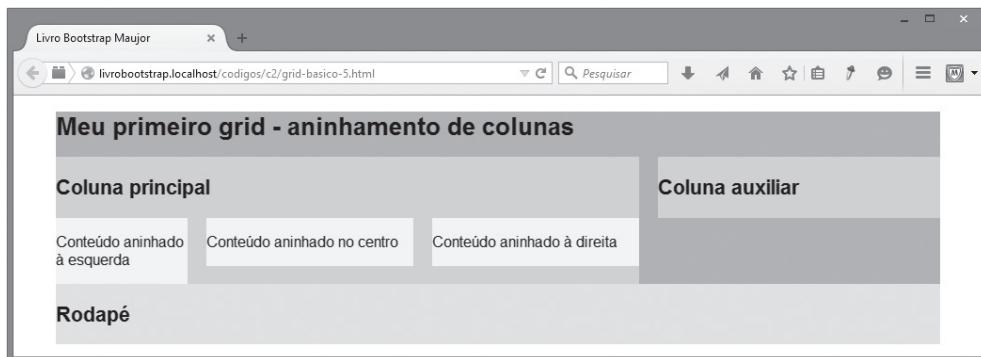
Figura 2.6 – Grid básico, aninhamento de conteúdos.

Note que, sendo o container dos três blocos de conteúdos a coluna principal que é formada pela união de oito células, temos disponível, para criar os blocos, estas oito células. Por essa razão, optamos, em nosso exemplo, pelo primeiro bloco, que foi resultado da união de duas células, e o segundo e o terceiro, de três células.

Assim, cabe a pergunta: e se precisarmos de três blocos com larguras iguais? Com oito células disponíveis, temos duas possibilidades para múltiplos de três: unir seis células duas a duas e distribuir as duas células restantes entre as três ou tomar uma célula três vezes distribuindo as cinco células restantes entre elas.

Em ambos os casos, teríamos que criar regras CSS personalizadas.

2.4.6 Semântica

Conforme foi visto, os conceitos da construção de layouts baseados em sistema de grid se fundamentam em uma marcação HTML com sua estrutura geral padronizada, na qual os nomes dos atributos `class` são predefinidos e servem de referência para estabelecer regras CSS destinadas a posicionar e formatar os diferentes elementos do layout.

Você deve estar se perguntando: O layout é todo construído com elementos `div`? E a semântica? A resposta é: desde que você preserve os atributos e seus valores predefinidos, use o elemento que você quiser em substituição ao elemento `div`. Mas lembre-se de que os elementos substitutos devem ser declarados no nível de bloco, como são os elementos `div` substituídos.

Observe o arquivo constante do item [2.4.7] reescrito de forma semântica:

- **HTML**

```
1.  <div class="container">
2.      <header>
3.          <h1>Meu primeiro grid - semântica</h1>
4.      </header>
5.      <div class="row">
6.          <section class="span8">
7.              <header>
8.                  <h2>Coluna principal</h2>
9.              </header>
10.                 <div class="row">
11.                     <div class="span2" style="background: #f5f5f5;">
                            <p>Conteúdo aninhado á esquerda</p></div>
12.                     <div class="span3" style="background: #f5f5f5;">
                            <p>Conteúdo aninhado no centro</p></div>
13.                     <div class="span3" style="background: #f5f5f5;">
                            <p>Conteúdo aninhado à direita</p></div>
14.                 </div> <!-- .row dos conteúdos aninhados-->
15.         </section> <!-- /.span8 coluna principal -->
16.         <aside class="span4"><h2>Coluna auxiliar</h2></aside>
17.     </div> <!-- /.row  das colunas -->
18.     <div class="row">
19.         <footer class="span12"><h2>Rodapé</h2></footer>
20.     </div> <!-- /.row  do rodapé -->
21. </div> <!-- /.container -->
```

[.../c2/grid-basico-6.html]

O fato é que em projetos complexos a solução semântica proposta neste simples exemplo será inviável, e a verdade é que autores têm procurado solucionar este problema propondo algumas alternativas, tais como a criação de nomes de classes semânticos a serem processados por SASS ou LESS e outras, mas não existe uma solução semântica a toda prova, e este é um problema ainda não solucionado, comum à grande maioria dos frameworks CSS.

As versões anteriores ao IE9 do Internet Explorer não reconhecem os novos elementos da HTML5, por isso é necessário usar um script para forçar aqueles navegadores a reconhecer e estilizar tais elementos. Quando for o caso, nos exemplos deste livro, lincaremos para um script SHIM hospedado no CDN do Google, como mostrado no código a seguir.

```
<head>
...
<!--[if lt IE 9]>
   <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
```

Dica: Existem outros scripts que cumprem a mesma função, tal como o Modernizr. Use o de sua preferência e conforme as suas necessidades.

2.4.7 Espaçando colunas

No grid do nosso exemplo, tal como no Bootstrap, o espaçamento-padrão entre colunas é de 20px, contudo existe uma funcionalidade que permite ao autor aumentar esse espaçamento, fazendo com que uma ou mais colunas sejam incorporadas para definir o espaçamento. Por exemplo: considerando as doze colunas iniciais do grid, é possível construir uma coluna à esquerda com sete células e outra à direita com três células e com um espaçamento entre elas de — 12 – (3 + 7) = 2 — duas células (180px). Na figura 2.7, há o diagrama que ilustra essa configuração.
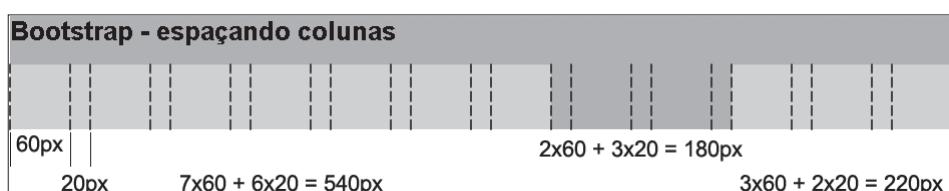
Figura 2.7 – Diagrama de espaçamento de colunas.

Quando usamos colunas para obter espaçamento, em ambas as extremidades do espaçamento é adicionado o espaçamento-padrão de 20px, daí, em nosso exemplo, no qual usamos duas colunas para espaçar, temos um espaçamento total de 2x60 + 3x20 = 180px, como mostrado na figura 2.7. Por outro lado, a união de colunas para formar uma coluna de largura maior não inclui os espaçamentos-padrão nas extremidades.

A marcação HTML e as regras de estilo específicas para criar o layout proposto no diagrama são mostradas a seguir.

- **HTML**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="utf-8">
<title>Livro Bootstrap Maujor</title>
<style type="text/css">
/* regras de estilo conforme exemplos anteriores */
.span3 { width: 220px; }
.span7 { width: 540px; }
.offset2 { margin-left: 180px; } /* classe define largura do espaçamento */

</style>
</head>
<body>
   <div class="container">
      <h1>Bootstrap - espaçando colunas</h1>
         <div class="row">
            <div class="span7">
               <h2>Coluna principal</h2>
            </div> <!-- .span7 -->
         <div class="span3 offset2">
            <h2>Coluna auxiliar</h2>
         </div> <!-- .span3 -->
      </div><!-- /.row  -->
   </div> <!-- /.container -->
</body>
</html>
```

[.../c2/espacando-colunas.html]

A renderização do arquivo anterior deve ficar conforme mostrado na figura 2.8.
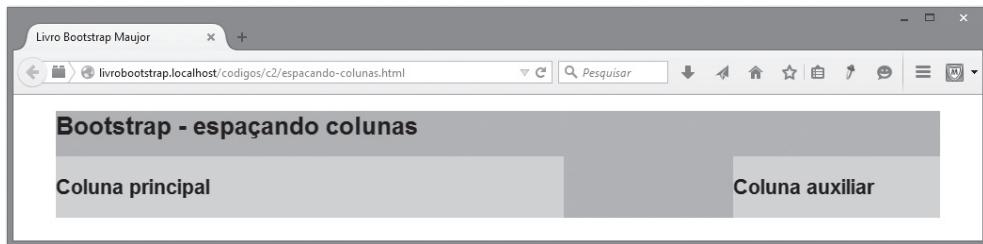
Figura 2.8 – Espaçando colunas.

Para unir colunas do grid e transformá-las em espaçamentos, o nosso exemplo definine o valor `offset*` para o atributo `class`. Esse valor termina com um número que define o número de colunas a unir para criar o espaçamento. Dessa forma, `offset3` une três colunas, `offset7` une sete colunas, e assim por diante. As regras CSS que se aplicam a esses valores de classe são mostradas a seguir.

- **CSS**

```
.offset1  { margin-left: 100px; }
.offset2  { margin-left: 180px; }
.offset3  { margin-left: 260px; }
.offset4  { margin-left: 340px; }
.offset5  { margin-left: 420px; }
.offset6  { margin-left: 500px; }
.offset7  { margin-left: 580px; }
.offset8  { margin-left: 660px; }
.offset9  { margin-left: 740px; }
.offset10 { margin-left: 820px; }
.offset11 { margin-left: 900px; }
```

Note que o espaçamento é definido atribuindo-se uma margem esquerda para a coluna. Assim, devemos declarar o valor `offset*` para o atributo `class` contido no elemento que cria a coluna à direita do espaçamento. Em nosso exemplo, o elemento `div` que cria a coluna auxiliar.

2.5 Grid fluido

Todos os exemplos mostrados até aqui foram criados de modo a obter um grid fixo com 940px de largura centrado na tela. Para obter um grid fluido basta declarar os valores das larguras em porcentagens.

2.6 Grid responsivo

Para obter um grid responsivo basta declarar uma metatag apropriada e criar regras CSS com uso das media queries.

2.7 Conclusão

Neste capítulo, estudamos os fundamentos de criação e funcionamento de grids CSS. O Bootstrap é um grid CSS bem mais complexo que o mostrado neste capítulo, mais os princípios aqui mostrados são válidos para o entendimento do seu funcionamento estrutural.

CAPÍTULO 3

CSS

Neste capítulo estudaremos os padrões básicos de estilização adotados pelo Bootstrap. Examinaremos as regras CSS padrão para o grid do Bootstrap, tipografia, códigos e blocos de códigos. Veremos com detalhes as diferentes formas de estilização de tabelas, as funcionalidades CSS para apresentação de formulários, a coleção de botões pré-estilizados e prontos para uso na interface a projetar. Estudaremos os três tipos de estilização para apresentação de imagens e para finalizar o capítulo veremos quais são e como utilizar os ícones padrão fornecidos pelo Bootstrap.

3.1 Grid

O grid é a espinha dorsal de qualquer framework CSS. É ele que determina o posicionamento de cada elemento em uma página, sendo assim usado para construir o layout. Um sólido conhecimento das marcações HTML padrão disponíveis para estruturar layout é o primeiro requisito para se aprender a usar o Bootstrap. Ao construir um layout, por mais complexo que ele seja, é necessário saber quais são os elementos, classes, atributos e estruturas de marcação do grid usar. E, tão importante e fundamental quanto os elementos HTML são os atributos a serem declarados na marcação, pois são eles que servirão de referência para que a folha de estilo padrão do Bootstrap aplique as regras CSS previstas para criar os diferentes componentes e widgets padrões do Bootstrap.

Portanto, desde já, a exemplo do que estudamos no capítulo 1, esteja ciente de que um sistema de grids se baseia em marcação *predefinida* e nomes de atributos, notadamente o atributo `class` também *predefinidos*.

3.1.1 Introdução

Uma aplicação desenvolvida com Bootstrap deve estar em conformidade com algumas premissas mandatórias conforme descritas a seguir.

- Estar contida em um elemento `div` container geral que recebe a classe de nome `container` conforme mostrado a seguir.

```
...
<body>
<div class="container">
  <!-- Aplicação criada com Bootstrap -->
</div> <!-- /.container -->
<body>
<html>
```

Opcionalmente, para obter um layout fluido (layout que preencha toda a largura da tela) o nome de classe `container` deve ser substituído por `container-fluid`. Não se preocupe com este nome, adiante neste livro, veremos como usar e quais são os efeitos de se usar, alternativamente, este nome.

3.1.2 Breakpoints

As CSS do Bootstrap foram desenvolvidas seguindo as premissas da filosofia "mobile first". Isto significa que as regras de estilo iniciais foram criadas para contemplar um layout de uma coluna no qual os blocos de conteúdos vão sendo posicionados na vertical um após outro e suas larguras ocupam todo o espaço disponível em dispositivos móveis, com viewports de largura menor do que 768px.

Para fins didáticos denominaremos, neste livro, as regras CSS para dispositivos com viewport menor do que 768px de *regras CSS iniciais*. Os criadores do Bootstrap estabeleceram que por padrão o primeiro breakpoint fosse em 768px.

A partir da largura de viewport igual a 768px, usando-se media queries foi criado um conjunto de regras CSS a serem aplicadas juntamente com as regras CSS iniciais.

Os criadores do Bootstrap estabeleceram que por padrão o segundo breakpoint fosse em 992px e usando-se media queries foi criado um conjunto de regras CSS a serem aplicadas juntamente com as regras CSS anteriores.

Os criadores do Bootstrap estabeleceram que por padrão o terceiro breakpoint fosse em 1200px e usando-se media queries foi criado um conjunto de regras CSS a serem aplicadas juntamente com as regras CSS anteriores para larguras de viewport maiores do que 1200px.

Assim, em um layout criado com uso do Bootstrap, por padrão, é possível que se defina até quatro acomodações do layout de acordo com a largura da viewport. Em design responsivo há um princípio que diz que quem determina os breakpoints é o layout e não a largura dos dispositivos. Assim sendo não se deve estabelecer breakpoints “a priori”.

O Bootstrap estabelece quatro breakpoints e na maioria dos casos eles atenderão seu layout. Se ajustes forem necessários com mudança, ou mesmo criação de novos breakpoints o autor poderá criar regras de estilo complementares ou mesmo usar LESS do Bootstrap para processar uma nova folha de estilo personalizada. Neste livro vamos desenvolver nossos exemplos considerando os quatro breakpoints mostrados.

Para cada faixa de comportamento do layout de acordo com os breakpoints está prevista uma largura máxima para o container geral da aplicação conforme mostrado a seguir.

- Até 768px a largura terá o valor CSS `auto`.
- Entre 768px e 992px a largura máxima é igual a 750px.
- Entre 992px e 1200px a largura máxima é igual a 970px.
- Acima de 1200px a largura máxima é igual a 1170px.

3.1.3 Linhas do grid

Criam-se linhas do grid com um elemento `div` que recebe a classe de nome `row` conforme mostrado a seguir, para criar duas linhas.

```
<div class="container">
   <div class="row"> <!-- Linha -->
      <!-- Colunas -->
   </div> <!-- /.row -->
   <div class="row"> <!-- Linha -->
      <!-- Colunas -->
   </div> <!-- /.row -->
/div> <!-- /. container -->
```

- Dentro do elemento `div` que cria uma linha (`div.row`) insere-se como elemento-filho elementos para criar colunas. Colunas, formadas por células ou constituída de uma célula somente, são criadas com uso de um elemento `div` que recebe um ou mais atributos e cujo atributo classe tem valor cujo formato geral é `col-xs-*`, `col-sm-*`, `col-md-*` e `col-lg-*`, sendo o sinal * um número de 1 a 12 que define a quantidade de células que formam a coluna. O número 12 é, por padrão, o número máximo de colunas em uma linha. É possível unir células (cada uma das 12 colunas de uma linha) com a finalidade de se obter colunas com largura maior. Uma linha pode conter qualquer combinação de colunas desde que o número máximo de células não ultrapasse a 12, podendo ser menor. A finalidade de cada um dos valores de classe para os `div` containers das células é conforme explicado a seguir.
  - `col-xs-*` – *Telefones*; define colunas para dispositivos com viewport extra pequena (menor que 768px).
  - `col-sm-*` – *Tablets*; define colunas para dispositivos com viewport pequena (de 768px até 991px).
  - `col-md-*` – *Desktop médio*; define colunas para dispositivos com viewport média (de 991px até 1199px).
  - `col-lg-*` – *Desktop largo*; define colunas para dispositivos com viewport larga (maior que 1199px).

Se for definido para uma linha um número de colunas cuja soma total de células ultrapasse a 12, cada grupo de colunas a mais será posicionado a seguir no fluxo, tal como, se houvesse uma "quebra de linha".

Caso contrário, se for definido para uma linha um número de colunas cuja soma total de células seja menor do que 12, o grupo de células em falta será posicionado à direita e não receberá conteúdos.

3.1.4 Colunas do grid

Conhecendo quais são os breakpoints padrão do Bootstrap vamos examinar novamente os valores de classes para definir colunas, mostrados no item anterior. Cada uma das quatro classes mostradas se destina a aplicar, as regras de estilo previstas para um breakpoint, no elemento ao qual ela classe for definida.

3.1.4.1 Fundamentos

O entendimento das premissas relacionadas a seguir é fundamental para uma perfeita compreensão e consequentemente uso correto e apropriado das classes destinadas à criação de colunas. Conforme mostrado anteriormente os valores do atributo classe do container das colunas são os seguintes: `col-xs-*`, `col-sm-*`, `col-md-*`, `col-lg-*`.

- Quaisquer que sejam os valores de classe definidos para um container de colunas, quando a largura da viewport for menor do que768px serão aplicadas, no container, as regras de estilo iniciais.
- Se não for definido um valor de classe para o container da coluna as regras de estilo aplicadas ao container serão as regras de estilo iniciais, ou seja, aquelas para viewport menor do que 768px (mobile-first) e o layout terá comportamento mobile em todas as larguras de viewport. Então, para que serve `col-xs-*`? Para sobrescrever o comportamento colunar em viewport menor do que 768px, permitindo, por exemplo, que se crie um layout de duas, ou mais colunas em larguras de viewport menores do que 768px.
- Se for definido somente o valor `col-sm-*` o container terá o comportamento colunar em larguras de viewport menores que 768px e outro em maiores.
- Se for definido somente o valor `col-md-*` o layout terá o comportamento colunar em larguras de viewport menores que 992px e outro em maiores.
- Se for definido somente o valor `col-lg-*` o layout terá o comportamento colunar em larguras de viewport menores que 1200px e outro em maiores.
- O efeito de se definir dois ou mais valores de classe faz com que o comportamento do layout varie de acordo com os valores de classe definidos e os respectivos breakpoints.

Vejamos alguns exemplos práticos que ilustram a criação de colunas em diferentes situações.

Exemplo 1

- **CSS** – Coloca uma cor de fundo e uma borda nas colunas com a finalidade de facilitar a visualização.

```
.row div[class^="col-"] {
   background: #fafafa;
   border:1px solid #ccc;
}
```

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-1">.col-md-1</div>
      <div class="col-md-1">.col-md-1</div>
      <!- repete 9 vezes -->
      <div class="col-md-1">.col-md-1</div>
   </div> <!-- /.row -->
   <div class="row">
      <div class="col-md-8">.col-md-8</div>
      <div class="col-md-4">.col-md-4</div>
   </div> <!-- /.row -->
   <div class="row">
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
      <div class="col-md-4">.col-md-4</div>
   </div> <!-- /.row -->
   <div class="row">
      <div class="col-md-6">.col-md-6</div>
      <div class="col-md-6">.col-md-6</div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/classes-para-criar-colunas.html]

Neste exemplo usou-se o valor de classe `col-md-*` para obter um layout destinado a dispositivos móveis (telefones e tablets) em viewport abaixo de 992px e outro para desktop em viewport acima de 992px.

Observe na figura 3.1 o resultado da aplicação da classe `col-md-*` em diferentes larguras de viewport. Mostramos na figura a largura máxima padrão definida pelo Bootstrap para o container geral da aplicação (`div.container`) para diferentes larguras de viewport.
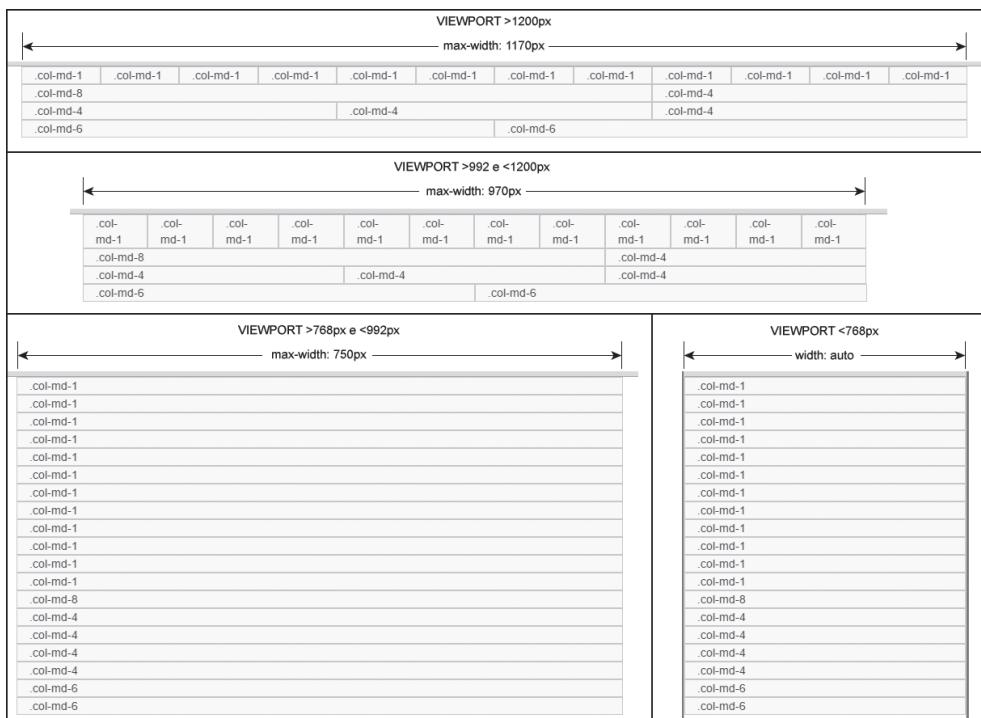
Figura 3.1 – Classes para criar colunas.

Esse exemplo prático demostra a aplicação de classe para criação de colunas e encontra-se disponível para consulta online.

3.1.4.2 Número máximo de colunas em uma linha

Sabemos que o número máximo de células em uma linha é 12 e já dissemos que se um número maior de células for declarado para uma linha, as células a mais serão posicionadas abaixo como se existisse uma “quebra de linha”. No exemplo a seguir mostramos a marcação para uma linha que comprova este comportamento.

- **CSS**

```
.row div[class^="col-"] {
   background: #fafafa;
   border:1px solid #ccc;
}
.row { border: 2px dotted black; }
```

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-xs-6">6</div>
      <div class="col-xs-4">4</div>
      <div class="col-xs-5">5</div>
      <div class="col-xs-3">3</div>
      <div class="col-xs-7">7</div>
   </div>
</div> <!-- /.container -->
```

[.../c3/numero-maximo-colunas-em-linha.html]

Neste exemplo definiu-se uma linha com um número total de células igual a 25 células.

Foi incorporada ao documento uma folha de estilo adicional para aplicar cor de fundo e borda em cada coluna e uma borda pontilhada na linha com a finalidade de facilitar a visualização do resultado. Observe na figura 3.2 a renderização da linha.

Figura 3.2 – Número máximo de colunas em uma linha.

Notar que a soma das células das três primeiras colunas é 6+4+5 = 15 > 12. Assim as duas primeiras colunas 6+4=10<12 são acomodadas e a terceira coluna vai para baixo.

A soma células das três últimas colunas é 5+3+7 = 15 > 12 e o processo se repete.

Notar que o espaço que “sobra” em cada fileira é igual ao número de células que faltam para completar 12 na fileira.

Esse exemplo prático demostra o comportamento das colunas quando se declara mais de 12 colunas em uma linha e encontra-se disponível para consulta online.

3.1.4.3 Espaçando colunas

O Bootstrap prevê valores para o atributo classe do tipo `col-xs-offset-*`, `col-sm-offset-*`, `col-md-offset-*`, `col-lg-offset-*` nos quais o sinal * é um número inteiro de 1 a 11. Estes valores se destinam a criar um espaçamento à esquerda da coluna. O número de células que define a largura do espaçamento é computado na soma do total de células que cabem em uma linha, que como já vimos é 12. Observe o exemplo a seguir que mostra o espaçamento entre colunas em três linhas diferentes.

- **CSS**

```
.row div[class^="col-"] {
   background: #fafafa;
   border:1px solid #ccc;
}
.row { margin-bottom: 5px; } /* separa linhas para facilitar visualização */
```

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-3">3</div>
      <div class="col-md-2 col-md-offset-3">2</div>
      <div class="col-md-1 col-md-offset-3">1</div>
   </div> <!-- /.row -->
   <div class="row">
      <div class="col-md-5 col-md-offset-2">5</div>
      <div class="col-md-3 col-md-offset-1">3</div>
   </div> <!-- /.row -->
   <div class="row">
      <div class="col-md-8 col-md-offset-2">8</div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/espacando-colunas.html]

Neste exemplo as três linhas foram definidas com espaçamento entre colunas conforme descrito a seguir.

Na primeira linha temos uma coluna com três células a seguir uma coluna com duas células com um espaçamento de três células à esquerda e a seguir uma coluna com uma célula com um espaçamento de três células à esquerda. A soma total de células é 12.

Na segunda linha temos uma coluna com cinco células com um espaçamento de duas células à esquerda, a seguir uma coluna com quatro células com um espaçamento de uma célula à esquerda. A soma total de células é 12.

Na terceira linha temos uma coluna com oito células com um espaçamento de duas células à esquerda perfazendo um total de 11 células na linha. Notar que o espaço de uma célula que falta para totalizar as 12 células na linha é posicionado a direita na linha.

Foi incorporada ao documento uma folha de estilo adicional para aplicar cor de fundo e borda em cada coluna e uma margem entre linhas com a finalidade de facilitar a visualização do resultado. Observe na figura 3.3 mos a renderização dos espaçamentos nas três linha.

Figura 3.3 – Espaçando colunas.

Esse exemplo prático demostra como criar espaçamento entre colunas e encontra-se disponível para consulta online.

3.1.4.4 Aninhando colunas

É válido aninhar colunas em tantos níveis quando se queira. Para aninhar colunas basta que se crie linhas dentro de colunas existentes. Neste caso as linhas criadas conterão colunas aninhadas na coluna onde a linha foi criada. No exemplo a seguir em uma linha com duas colunas, uma com oito células e outra com quatro células aninhou-se na primeira coluna uma linha com 3 células e na segunda uma com duas células.

- **CSS**

```
.row div[class^="col-"] {
   background: #fafafa;
   border:1px solid #ccc;
   font-size: 30px;
   font-weight: bold;
   padding-top: 10px;
   padding-bottom: 10px;
```

```
}
.row .row > div[class^="col-"] {
   background: #f1f1f1;
   border:1px solid #black;
   font-size: 20px;
   font-weight: bold;
   }
```

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-8">
      8
         <div class="row">
            <div class="col-md-2">2</div>
            <div class="col-md-7">7</div>
            <div class="col-md-3">3</div>
         </div> <!-- /.row -->
      </div>
      <div class="col-md-4">
      4
         <div class="row">
            <div class="col-md-6">6</div>
            <div class="col-md-6">6</div>
         </div> <!-- /.row -->
      </div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/aninhando-colunas.html]

Neste exemplo as três linhas foram definidas com espaçamento entre colunas.

Foi incorporada ao documento uma folha de estilo adicional para aplicar cor de fundo e borda em cada coluna com a finalidade de facilitar a visualização do resultado.

Observe na figura 3.4 a renderização das colunas aninhadas.

Esse exemplo prático demostra como aninhar colunas e encontra-se disponível para consulta online.

Figura 3.4 – Aninhando colunas.

Dica: Notar que para o grid do Bootstrap o aninhamento de colunas se faz de modo a que o número máximo de colunas aninhadas seja sempre igual a doze, independentemente do nível de aninhamento, pois a largura das células é definida em porcentagem.

3.1.4.5 Ordenando colunas

O Bootstrap prevê os valores de classe `col-xs-push-*`, `col-sm-push-*`, `col-md-push-*`, `col-lg-push-*` que se destinam a "empurrar" a coluna para a direita e os valores `col-xs-pull-*`, `col-sm-pull-*`, `col-md-pull-*`, `col-lg-pull-*` que se destinam a "puxar" a coluna para a esquerda fazendo com que elas troquem de posição, ou seja, alterando sua ordem. Observe o exemplo mostrado a seguir que esclarece o uso destas classes.

- **CSS**

```
.row div[class^="col-"] {
  background: #fafafa;
  border:1px solid #ccc;
  font-size: 30px;
  font-weight: bold;
  padding: 10px 0;
  }
```

- **HTML**

```
<div class="container">
  <div class="row">
    <div class="col-md-5 col-md-push-7">5</div>
    <div class="col-md-7 col-md-pull-5">7</div>
  </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/ordenando-colunas.html]

Neste exemplo, na marcação HTML, a coluna 5 está à esquerda e a coluna 7 à direita e na renderização há troca das posições.

Esse exemplo prático demostra como ordenar colunas e encontra-se disponível para consulta online.

3.1.5 Container fluido

Vimos no item anterior que para cada faixa de largura de viewport foi estabelecida uma largura máxima para o container geral da aplicação. Este comportamento pode ser alterado de modo a que em qualquer largura de viewport o layout se comporte como fluido, ou seja, expanda para ocupar toda a largura disponível. Para isso basta que o valor da classe do elemento `div` container geral seja alterado de `container` para `container-fluid` como mostrado a seguir.

- **HTML**

```
<div class="container container-fluid">
   <!-- igual ao exemplo mostrado na figura 3.1 -->
</div> <!-- /.container-fluid -->
```

[.../c3/container-fluido.html]

Neste exemplo mostramos como alterar o exemplo mostrado na figura 3.1 para que se comporte como layout fluido.

Esse exemplo prático demostra como tornar o layout fluido e encontra-se disponível para consulta online.

É válido usar-se os dois tipos de container (`.container` e `.container-fluid`) em uma mesma página. Um exemplo comum é aquele no qual o site adota uma faixa que sempre se estende por toda a largura da viewport (faixa fluída) para o topo e outra para o rodapé enquanto os conteúdos, em telas maiores, são renderizados em uma largura menor que a da viewport e centralizados na tela.

A marcação HTML mostrada a seguir esclarece a estrutura geral para esta situação.

- **HTML**

```
<header class="container-fluid">TOPO</header>
<main class="container">CONTEÚDOS</main>
<footer class="container-fluid"></footer>
```

3.2 Tipografia

Antes de começar a desenvolver layouts, é necessário aprofundar os conhecimentos que já adquirimos nos capítulos anteriores, com a finalidade de conhecer e saber usar as poderosas ferramentas que temos em mãos quando optamos pelo uso do Bootstrap. Vamos falar de estilização dos elementos tipográficos de uma aplicação web. Servirá de base para nossos estudos de estilização, o arquivo *bootstrap.css*.

Alerta: Os valores das propriedades CSS mostrados neste capítulo são valores padrão do Bootstrap. Caso alguns, ou muitos, desses valores não seja o mais indicado para o seu projeto, não se preocupe, pois você não precisará editar e alterar manualmente a folha de estilo. Você poderá baixar uma versão personalizada da folha de estilo em *http://getbootstrap.com/customize/*.

3.2.1 Estilização do elemento body

As declarações CSS para o elemento `body` são mostradas a seguir.

- **CSS**

```
body {
   margin: 0;
   font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
   font-size: 14px;
   line-height: 1.42857143;
   color: #333;
   background-color: #fff;
   }
```

A família de fontes para todos os conteúdos textuais dos elementos do Bootstrap, exceto para os elementos `code`, `kbd`, `pre` e `sample` é definida no seletor `body`.

O tamanho de fonte padrão é 14px com a propriedade `line-height` definida em 1.42857143 que para um tamanho de fonte de 14px resulta em 1.42857143 x 14 = 20px. A cor dos textos é a preta e do fundo a branca.

3.2.2 Estilização dos elementos h1 - h6

Os tamanhos de fonte para os níveis de cabeçalhos são conforme mostrados a seguir.

```
h1, .h1 { font-size: 36px; }
h2, .h2 { font-size: 30px; }
h3, .h3 { font-size: 24px; }
h4, .h4 { font-size: 18px; }
h5, .h5 { font-size: 14px; }
h6, .h6 { font-size: 12px; }
```

Notar que está prevista uma classe com o nome igual ao nome da tag de cabeçalho e se destina a uso geral.

O elemento `small` quando usado dentro de um elemento de cabeçalho recebe uma estilização especial em cor cinza clara e tamanho de fonte 80% do tamanho de fonte do cabeçalho. Opcionalmente pode-se usar a classe `small` para obter o mesmo efeito.

O exemplo mostrado a seguir esclarece esta estilização para cabeçalhos.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-12">
         <h1>h1 - Cabeçalho nível 1 <small>Texto small</small></h1>
         <h2>h2 - Cabeçalho nível 2 <small>Texto small</small></h2>
         <h3>h3 - Cabeçalho nível 3 <small>Texto small</small></h3>
         <h4>h4 - Cabeçalho nível 4 <small>Texto small</small></h4>
         <h5>h5 - Cabeçalho nível 5 <small>Texto small</small></h5>
         <h6>h6 - Cabeçalho nível 6 <small>Texto small</small></h6>
      </div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/estilização-de-cabecalhos.html]

Observe na figura 3.5 o resultado da aplicação da estilização padrão em elementos de cabeçalho, bem como o efeito do elemento `small`.

h1 - Cabeçalho nível 1 Texto small
h2 - Cabeçalho nível 2 Texto small
h3 - Cabeçalho nível 3 Texto small
h4 - Cabeçalho nível 4 Texto small
h5 - Cabeçalho nível 5 Texto small
h6 - Cabeçalho nível 6 Texto small
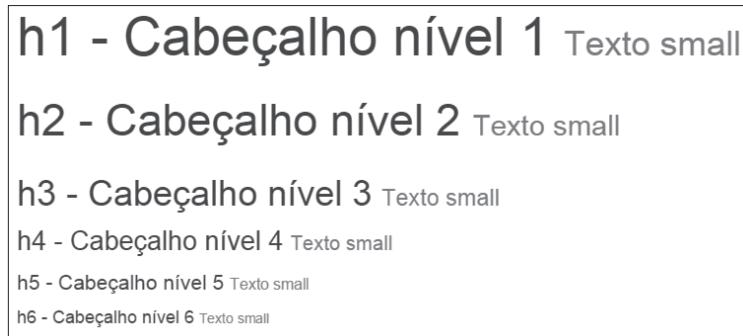
Figura 3.5 – Estilização de cabeçalhos.

Esse exemplo prático demostra a estilização padrão de elementos para cabeçalhos e encontra-se disponível para consulta online.

3.2.3 Estilização de elementos para textos inline

Os elementos `mark`, `del`, `ins`, `s`, `u`, `em`, `strong`, `i`, `b` e `small` são usados e estilizados conforme previstos nas especificações para a HTML5.

O exemplo mostrado a seguir esclarece a estilização de elementos para textos inline.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-12">
         <mark>Estilização do elemento></mark> <code>mark</code<hr>
         <del>Estilização do elemento</del> <code>del</code><hr>
         <ins>Estilização do elemento</ins> <code>ins</code><hr>
         <s>Estilização do elemento</s> <code>s</code><hr>
         <u>Estilização do elemento</u> <code>u</code><hr>
         <em>Estilização do elemento</em> <code>em</code><hr>
         <strong>Estilização do elemento</strong> <code>strong</code><hr>
         <i>Estilização do elemento</i> <code>i</code><hr>
         <b>Estilização do elemento <code>b</code></b><hr>
         <small>Estilização do elemento <code>small</code></small>
      </div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/estilizacao-de-elementos-para-textos-inline.html]

Observe na figura 3.6 o resultado da aplicação da estilização padrão em elementos para textos inline.

Figura 3.6 – Estilização de elementos para textos inline.

Esse exemplo prático demostra a estilização padrão de elementos para textos inline e encontra-se disponível para consulta online.

3.2.4 Estilização do elemento abbr

O elemento `abbr` é usado para marcar abreviaturas em geral e estilizado de modo a mostrar uma janela tooltip com o significado da abreviatura quando o usuário coloca o ponteiro do mouse sobre ela. O texto do tooltip é definido no atributo `title` do elemento `abbr`.

Além desse comportamento, o Bootstrap prevê ainda o valor `initialism` para o atributo classe do elemento `abbr`. Definir esse valor de classe no elemento `abbr` faz com que a abreviatura seja renderizada com um tamanho de fonte igual a 90% do tamanho de fonte padrão da abreviatura.

O exemplo mostrado a seguir esclarece a estilização deste elemento.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-12">
```

```
          <p><abbr title="HyperTextMarkupLanguage">HTML</abbr> é a
          linguagem de marcação para aplicações web.</p>
          <p><abbr title="HyperTextMarkupLanguage" class="initialism">HTML</abbr> é a
          linguagem de marcação para aplicações web.</p>
      </div>
    </div> <!-- /.row -->
    </div> <!-- /.container -->
</body>
</html>
```

[.../c3/estilizacao-do-elemento-abbr.html]

Esse exemplo prático demostra a estilização padrão do elemento `abbr` e a estilização com uso da classe `initialism` e encontra-se disponível para consulta online.

3.2.5 Estilização do elemento address

O elemento `address`, introduzido nas especificações pela HTML5 é usado para marcar informações de contato relacionadas aos conteúdos de um elemento `article` ou ao conteúdo do elemento `body` como um todo. Não deve ser usado para marcar endereço postal generalizadamente exceto nos casos em que o endereço postal forneça informações de contato relacionadas aos conteúdos citados.

Você pode visualizar os efeitos de estilização padrão do Bootstrap para o elemento `address` consultando o arquivo e mostrado a seguir.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-12">
         <p>Contato com o autor do artigo:</p>
         <address>
            <strong>Maurício Samy Silva</strong><br>
            Rua: General Paulino 123 sala 4567<br>
            <abbr title="Código de endereçamento postal">CEP</abbr>: 00000-000<br>
Carapena, AM.
         </address>
      </div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/estilizacao-do-elemento-address.html]

Esse exemplo prático demostra a estilização padrão do elemento `address` e encontra-se disponível para consulta online ou download e consulta local.

3.2.6 Estilização dos elementos blockquote e cite

O elemento `blockquote` é usado para marcar um bloco de texto, ou seção, contendo uma citação extraída de uma fonte externa ao documento. O elemento `cite` destina-se a marcar o título de um trabalho, ou obra, por exemplo: o título de um livro, um filme, um programa de TV, um ensaio, um poema, uma música, um jogo etc.

Não confundir o elemento `cite` com o atributo `cite`, esse deve ser usado com os elementos `blockquote` e `q` e cujo valor deve ser um URL apontando para o endereço web de onde foi extraída a citação marcada por aqueles elementos.

Você pode visualizar os efeitos de estilização padrão do Bootstrap para o elemento `blockquote` e para o uso do valor `blockquote-reverse` para o atributo `class` consultando o arquivo mostrado a seguir.

- **HTML**

```
<!DOCTYPE html>
   <div class="container">
      <div class="row">
      <div class="col-md-12">
         <blockquote cite="http://www.editoranovatec.com.br...">
            <p>A tentativa de o W3C evoluir...</p>
         <footer>
            <cite title="HTML5 A linguagem de marcação que revolucionou a web">Maujor,
               ...</cite>
         </footer>
         </blockquote>
      </div>
      </div> <!-- /.row -->
      <hr>
      <div class="row">
      <div class="col-md-12">
         <blockquote cite="http://www.editoranovatec.com.br..." class="blockquote-reverse">
            <p>A tentativa de o W3C evolui...</p>
         <footer>
            <cite title="HTML5 A linguagem de marcação que revolucionou a web">Maujor,
               ...</cite>
```

```
            </footer>
            </blockquote>
        </div>
        </div> <!-- /.row -->
    </div> <!-- /.container -->
```

[.../c3/estilizacao-do-elemento-blockquote.html]

A renderização do arquivo anterior é conforme mostrado na figura 3.7. Observe o efeito da estilização padrão do Bootstrap para os elementos `blockquote` e `cite` e o efeito do atributo classe com valor `blockquote-reverse` destinado a alinhar o conteúdo de `blockquote` à direita.

A tentativa de o W3C evoluir a HTML para conformidade com XML fracassou, porque as páginas XHTML servidas como XML que contenham qualquer violação com a sintaxe XML causarão um erro fatal de parseamento e a página simplesmente não será renderizada.

— *Maujor, no seu livro HTML5 A linguagem de marcação que revolucionou a web*

A tentativa de o W3C evoluir a HTML para conformidade com XML fracassou, porque as páginas XHTML servidas como XML que contenham qualquer violação com a sintaxe XML causarão um erro fatal de parseamento e a página simplesmente não será renderizada.

Maujor, no seu livro HTML5 A linguagem de marcação que revolucionou a web —

Figura 3.7– Estilização dos elementos blockquote e cite.

3.2.7 Estilização de elementos de lista

Os elementos `ul`, `ol` e `dl` destinados a marcar listas receberam uma estilização padrão do Bootstrap cabendo destacar o valor `list-unstyled` para o atributo `class` dos elementos `ul` e `ol`. Definir esse valor de classe tem o efeito semelhante ao de se declarar `list-style: none;` nas CSS, ou seja, retira os marcadores das listas.

O Bootstrap prevê o valor `list-inline` para o atributo classe dos elementos `ul` e `ol` que marcam as listas de ordenadas e não ordenadas. Definir esse valor de classe tem o efeito de alinhar horizontalmente os itens da lista.

O Bootstrap prevê também o valor `dl-horizontal` para o atributo `class` dos elementos `dl` que marcam as listas de definição. Definir esse valor de classe tem o efeito de alinhar horizontalmente os termos de definição (`dt`) com seus termos de descrição (`dd`).

Você pode visualizar os efeitos de estilização padrão do Bootstrap para os elementos de listas e para o uso dos valores `unistyled` e `dl-horizontal` para o atributo `class` consultando o arquivo mostrado a seguir.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-4">
         <h3>Lista não ordenada</h3>
         <ul>
            <li>Abacaxi</li>
            <li>Laranja
               <ul>
                  <li>Lima</li>
                  <li>Bahia</li>
               </ul>
            </li>
            <li>Maçã</li>
         </ul>
      </div>
      <div class="col-md-4">
         <h3>Lista ordenada</h3>
         <ol>
            <li>Abacaxi</li>
            <li>Laranja
               <ul>
                  <li>Lima</li>
                  <li>Bahia</li>
               </ul>
            </li>
            <li>Maçã</li>
         </ol>
      </div>
      <div class="col-md-4">
         <h3>Lista sem marcadores</h3>
```

```
        <ul class="list-unstyled">
          <li>Abacaxi</li>
          <li>Laranja
            <ul>
              <li>Lima</li>
              <li>Bahia</li>
            </ul>
          </li>
          <li>Maçã</li>
        </ul>
      </div>
    </div> <!-- /.row -->
    <div class="row">
      <div class="col-md-12">
        <h3>Lista não ordenada inline</h3>
          <ul class="list-inline">
            <li>Abacaxi</li>
            <li>Laranja</li>
            <li>Lima</li>
            <li>Bahia</li>
            <li>Maçã</li>
</ul>
      </div>
    </div> <!-- /.row -->
    <div class="row">
      <div class="col-md-6">
        <h3>Lista de definição padrão</h3>
          <dl>
            <dt>Abacaxi</dt>
              <dd>Ananás ou abacaxi...</dd>
            <dt>Laranja</dt>
              <dd>A laranja é o fruto...</dd>
            <dt>Maçã</dt>
              <dd>A maçã é o fruto pomáceo...</dd>
          </dl>
      </div>
      <div class="col-md-6">
        <h3>Lista de definição horizontal</h3>
          <dl class="dl-horizontal">
```

```
                <dt>Abacaxi</dt>
                    <dd>Ananás ou abacaxi é uma...</dd>
                <dt>Laranja</dt>
                    <dd>A laranja é o fruto produzido...</dd>
                <dt>Maçã</dt>
                    <dd>A maçã é o fruto pomáceo...</dd>
            </dl>
        </div>
    </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/estilizacao-de-listas.html]

A renderização do arquivo anterior é conforme mostrado na figura 3.8. Observe o efeito da estilização padrão do Bootstrap para os elementos de lista e para o atributo classe com os valores `list-unstyled` e `dl-horizontal`.

Figura 3.8 – Estilização de listas.

3.3 Códigos

Os elementos `code`, `kbd`, `var`, `samp` e `pre` destinados a marcar códigos em linha e em nível de blocos respectivamente receberam estilização padrão do Bootstrap cabendo destacar o valor `pre-scrollable` para o atributo `class` do elemento `pre`. Definir esse valor de classe tem o efeito de limitar a altura total do bloco de código a 350px provocando o aparecimento de barra de rolagem vertical quando a altura do bloco de código for maior que os 350px.

- O elemento `kbd` destina-se a marcar entrada do usuário, por exemplo teclas ou comandos de voz.
- O elemento `var` destina-se a marcar variáveis tanto em expressões matemáticas como variáveis de programas e scripts.
- O elemento `samp` destina-se a marcar saídas ou resultados de programas.
- O elemento `code` destina-se a marcar blocos de código.
- O elemento `pre` destina-se a marcar textos pré formatados.

Você pode visualizar os efeitos de estilização padrão do Bootstrap para os elementos de marcação de códigos e para o uso do valor `pre-scrollable` para o atributo classe consultando o arquivo mostrado a seguir.

- **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-3">
         <h3>Elemento &lt;code&gt;</h3>
            <p>Os elementos <code>&lt;div&gt;</code> e <code>&lt;span&gt;
               </code>não têm valor semântico.</p>
      </div>
      <div class="col-md-3">
         <h3>Elemento &lt;kbd&gt;</h3>
            <p>Para aumentar pressione <kbd>Ctr +</kbd></p>
      </div>
      <div class="col-md-3">
         <h3>Elemento &lt;var&gt;</h3>
            <p>A equação da reta é <var>y</var> = a<var>x</var> + b</p>
      </div>
      <div class="col-md-3">
         <h3>Elemento &lt;samp&gt;</h3>
            <p>Os retorno do script foi <samp>true</samp></p>
      </div>
   </div> <!-- /.row -->
<hr>
   <div class="row">
      <div class="col-md-6">
      <h3>Elemento &lt;pre&gt;</code></h3>
      <pre>
         &lt;!DOCTYPE html>
```

```
            &lt;html lang="pt-br">
            &lt;head>
            ...
         </pre>
         </div>
         <div class="col-md-6">
         <h3>Elemento &lt;pre class="pre-scrollable"&gt;</code></h3>
         <pre class="pre-scrollable">
            &lt;!DOCTYPE html>
            &lt;html lang="pt-br">
            &lt;head>
            ...
         </pre>
         </div>
      </div> <!-- /.row -->
   </div> <!-- /.container -->
```

[.../c3/estilizacao-de-elementos-para-codigo.html]

A renderização do arquivo anterior é conforme mostrado na figura 3.9.

Figura 3.9 – Estilização de elementos para marcar código.

Observe o efeito da estilização padrão do Bootstrap para os elementos para marcar códigos e para o atributo classe com o valor `pre-scrollable`. Notar que a estilização padrão prevê um fundo cinza e uma borda arredondada para ambos os elementos.

3.4 Tabelas

As funcionalidades do Bootstrap para estilização de tabelas baseiam-se em valores do atributo `class` a serem declarados no elemento `table` conforme descritos a seguir.

- `table` – Estilização básica da tabela compreendendo declaração de `padding` para as células e bordas horizontais entre linhas. Uso: `<table class="table">`.
- `table table-striped` – Estilização básica mais efeito zebra nas linhas. Usa o seletor `nth-child` e por isso não funciona nos IE8 e anteriores. Uso: `<table class="table table-striped">`.
- `table table-bordered` – Estiliza a tabela com a declaração de `padding` para as células e bordas. Uso: `<table class="table table-bordered">`.
- `table table-hover` – Estilização básica da tabela e efeito over nas linhas dentro de `tbody`. Uso: `<table class="table table-hover">`.
- `table table-condensed` – Estilização básica da tabela com linhas de altura reduzida. Uso: `<table class="table table-condensed">`.

São previstos ainda, valores do atributo `class` a serem declarados no elemento `tr`. Os valores (nomes) dessas classes e suas finalidades são listados a seguir.

- `active` – Estiliza a linha da tabela na cor azul cinza claro e indica uma linha com status de ativa. Uso: `<tr class="active">`.
- `success` – Estiliza a linha da tabela na cor verde clara e indica uma linha com status de ação bem sucedida. Uso: `<tr class="success">`.
- `info` – Estiliza a linha da tabela na cor azul clara e indica uma linha com status informativo. Uso: `<tr class="info">`.
- `warning` – Estiliza a linha da tabela na cor laranja clara e indica uma linha com status de alerta. Uso: `<tr class="warning">`.
- `danger` – Estiliza a linha da tabela na cor salmon clara e indica uma linha com status de perigo. Uso: `<tr class="danger">`.

Para demonstrar o uso e efeitos das classes de estilização de tabelas conforme descritas anteriormente usaremos nos exemplos que seguem a marcação de uma tabela conforme mostrada a seguir.

- **HTML**

```
<table>
<caption>Estoque de frutas</caption>
<thead>
   <tr>
      <th scope="col">No.</th>
      <th scope="col">Fruta</th>
      <th scope="col">Estoque<br>(kg)</th>
      <th scope="col">Preço/caixa<br>(R$)</th>
   </tr>
</thead>
<tfoot>
   <tr><td colspan="4">Valores finais para o mês de fevereiro/2012</td></tr>
</tfoot>
<tbody>
   <tr>
      <td>1</td>
      <td>Abacaxi</td>
      <td>270</td>
      <td>125,90</td>
   </tr>
   <!-- mais três linhas de frutas -->
</tbody>
</table>
```

3.4.1 Estilização básica

- **HTML**

```
<table class="table">
   <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-basica-de-tabela.html]

Este exemplo de estilização de tabela encontra-se disponível para consulta online ou download e consulta local.

Observe na figura 3.10 a renderização da tabela básica.

Tabela básica

<table class="table">

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40

Valores finais para o mês de fevereiro/2012

Figura 3.10 – Estilização de tabela básica.

3.4.2 Estilização com efeito zebra

- **HTML**

```
<table class="table table-striped">
  <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-de-tabela-efeito-zebra.html]

Este exemplo de estilização de tabela encontra-se disponível, para consulta online ou download e consulta local.

Observe na figura 3.11 a renderização da tabela com efeito zebra.

Para obter o efeito zebra o Bootstrap usa como seletor a pseudo-classe `:nth-child()` que não é suportada pelo Internet Explorer 8.

Tabela com efeito zebra

`<table class="table table-striped">`

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40

Valores finais para o mês de fevereiro/2012

Figura 3.11 – Estilização de tabela com efeito zebra.

3.4.3 Estilização com classes de contexto

Em tabelas as classes de contexto se destinam a estilizar as linhas da tabela com cores indicativas de um estado. Estas classes devem ser definidas para o elemento `tr`. As classes de contexto para tabelas e seus nomes e cores são: `active` (cor do tema), `success` (cor verde), `info` (cor azul), `warning` (cor amarela) e `danger` (cor salmão).

- **HTML**

```
<table class="table">
   <tr class="succes">...</tr>
   <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-de-tabela-com-classes-de-contexto.html]

Este exemplo de estilização de tabela encontra-se disponível, no site do livro, para consulta online ou download e consulta local.

3.4.4 Estilização com bordas

- **HTML**

```
<table class="table table-bordered">
   <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-de-tabela-com-bordas.html]

Este exemplo de estilização de tabela encontra-se disponível, no site do livro, para consulta online ou download e consulta local.

Observe na figura 3.12 a renderização da tabela com bordas.

Tabela com bordas

`<table class="table table-bordered">`

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40
Valores finais para o mês de fevereiro/2012			

Figura 3.12 – Estilização de tabela com bordas.

3.4.5 Estilização com efeito hover nas linhas

- **HTML**

```
<table class="table table-hover">
   <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-de-tabela-efeito-hover.html]

Este exemplo de estilização de tabela encontra-se disponível, no site do livro, para consulta online ou download e consulta local.

Observe na figura 3.13 a renderização da tabela com efeito hover nas linhas (o mouse está sobre a segunda linha).

Tabela com efeito hover

`<table class="table table-hover">`

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40

Valores finais para o mês de fevereiro/2012

Figura 3.13 – Efeito hover em linhas de tabela.

3.4.6 Estilização com linhas condensadas

- **HTML**

```
<table class="table table-condensed">
  <!-- marcação da tabela -->
</table>
```

[.../c3/estilizacao-de-tabela-condensada.html]]

Este exemplo de estilização de tabela encontra-se disponível, no site do livro, para consulta online ou download e consulta local.

Observe na figura 3.14 a renderização da tabela com linhas condensadas.

Tabela básica com efeito condensado

`<table class="table table-condensed">`

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40

Valores finais para o mês de fevereiro/2012

Figura 3.14 – Estilização de tabela com linhas condensadas.

Para obter este efeito simplesmente diminuiu-se o valor de `padding` das linhas.

3.4.7 Estilização responsiva

Por padrão as tabelas tem um comportamento igual ao comportamento dos demais elementos da marcação, ou seja, contraem para se adaptar à largura da viewport. O Bootstrap prevê a classe `table-responsive` que se destina a eliminar o comportamento de contração da tabela em viewports com largura inferior a 768px e criar uma barra de rolagem horizontal para a tabela. Notar que a tabela recebe um elemento `div` como container e é esse elemento que recebe a classe `table-responsive`.

- **HTML**

```
<div class="table-responsive">
   <table class="table">
      <!-- marcação da tabela -->
   </table>
</div
```

[.../c3/estilizacao-de-tabela-responsiva.html]

Este exemplo de estilização de tabela encontra-se disponível, no site do livro, para consulta online ou download e consulta local.

Observe na figura 3.15 a renderização da tabela com efeito responsivo.

Tabela com efeito responsivo

<div class="table responsive">...</div>

Estoque de frutas

No.	Fruta	Estoque (kg)	Preço/caixa (R$)
1	Abacaxi	270	125,90
2	Laranja	460	156,75
3	Maçã	150	98,20
4	Pera	190	74,40

Valores finais para o mês de fevereiro/201

Figura 3.15 – Estilização de tabela responsiva.

3.5 Formulários

As funcionalidades do Bootstrap para estilização de formulários baseiam-se em valores do atributo classe a serem declarados nos elementos de formulário e cujos seletores CSS e respectivas finalidades estudaremos a seguir.

3.5.1 Estilização de controles

O Bootstrap prevê várias classes a serem aplicadas nos diferentes controles de formulários possibilitando uma estilização padrão. Nesse item veremos como aplicar as classes de estilização de controles e mostraremos os efeitos dessa estilização.

3.5.1.1 Estilização de input e textarea

Os controles `input` tipo `text` e `textarea` recebem uma estilização padrão sem necessidade de declarar uma classe. A estilização consiste na aplicação de uma borda cinza com cantos arredondados. Ao ser dado o foco ao controle a borda assume a cor azul com um efeito de sombra em volta, também na cor azul.

Alerta: Os navegadores Internet Explorer versões anteriores ao IE9 não suportam bordas arredondadas nem o efeito de sombra para o foco, contudo nenhuma funcionalidade do controle se perde para aqueles navegadores, sendo que o foco no controle causa `outline` pontilhada.

A marcação HTML mínima para mostrar a renderização estilizada desses controles é conforme a seguir.

- **HTML**

```
<input type="text" placeholder="Nome">
<textarea rows="5" placeholder="Sua mensagem aqui"></textarea>
```

Na figura 3.16 mostramos a renderização padrão e o efeito ao ser dado o foco no controle.

Figura 3.16 – Estilização de caixas de texto.

3.5.1.2 Estilização de botões

O Bootstrap prevê classes para estilizar botões. Essas classes destinam-se a definir cores e dimensões dos botões. Teoricamente as classes podem ser aplicadas a qualquer elemento HTML da marcação, contudo a sua finalidade principal é para estilização dos elementos HTML para marcar âncoras `a` e botões, mais especificamente os elementos `button` e os elementos `input` dos tipos `button`, `submit` e `reset`.

Está prevista a classe `btn` que serve de seletor para regras CSS que estilizam o botão sem bordas, cor de fundo cinza e cantos arredondados.

Deve-se usar a classe `btn-*` em conjunto com a classe `btn` para completar a estilização do botão conforme descrito adiante.

O Bootstrap prevê oito classes para estilizar botões de forma geral. Os nomes dessas classes e suas finalidades são listadas a seguir.

- `btn` – Destina-se a estilização padrão de um botão. Estiliza o botão sem bordas, cor de fundo cinza e cantos arredondados.
- `btn-default` – Botão com fundo na cor branca e borda cinza arredondada.
- `btn-primary` – Botão com fundo na cor azul escura e borda arredondada.
- `btn-success` – Botão com fundo na cor verde clara e borda arredondada.
- `btn-info` – Botão com fundo na cor azul clara e borda arredondada.
- `btn-warning` – Botão com fundo na cor laranja clara e borda arredondada.
- `btn-danger` – Botão com fundo na cor salmão clara e borda arredondada.
- `btn-link` – Esta classe se destina a ser aplicada (em conjunto com a classe `btn`) exclusivamente aos botões marcados com o elemento button. Ela transforma o botão padrão marcado com aquele elemento em um link padrão (como se fosse marcado com o elemento `a`).

Ao ser dado o foco a qualquer um dos botões o fundo assume uma cor mais escura.

Embora o elemento `a` destinado a marcar âncoras em geral, não seja um controle de formulário, pode ser estilizado como um botão com uso destas classes.

Alerta: Os navegadores Internet Explorer versões anteriores ao IE9 não suportam bordas arredondadas nem o efeito degradê que, nesses casos, se transforma em uma cor cinza sólida, contudo nenhuma funcionalidade dos botões se perde para aqueles navegadores.

A marcação HTML mínima para mostrar a renderização estilizada de alguns botões é conforme a seguir.

- **HTML**

```
<button type="button" class="btn btn-primary">button primary</button>
<button type="button" class="btn btn-default">button default</button>
<input type="submit" class="btn btn-success" value="input-submit success">
<input type="button" class="btn btn-warning" value="input-button warning">
<a href="#" class="btn btn-dange
<button type="button" class="btn btn-link">button link</button>
```

[.../c3/estilizacao-de-botoes.html]

Ao usar um elemento `a` para criar um botão não esqueça de marcar o atributo e seu valor `role="button"`.

Na figura 3.17 mostramos a renderização dos botões conforme a marcação anterior.

Figura 3.17 – Estilização de botões.

Por padrão, os botões são estilizados com um `padding` que determina suas dimensões. As classes `.bt-lg`, `.btn-sm` e `.btn-xs` possibilitam que se altere aquele valor padrão de `padding` e se obtenha botões com dimensões maiores, menores e muito menores que o botão padrão respectivamente.

A marcação HTML para criar tais botões é conforme a seguir.

- **HTML**

```
<button type="button" class="btn btn-active btn-lg">botão maior</button>
<button type="button" class="btn btn-active">botão padrão</button>
<button type="button" class="btn btn-active btn-sm">e="button"
   class="btn btn-active btn-xs">botão muito menor</button>
```

[.../c3/dimensoes-de-botoes.html]

Na figura 3.18 mostramos a renderização dos botões conforme a marcação anterior.
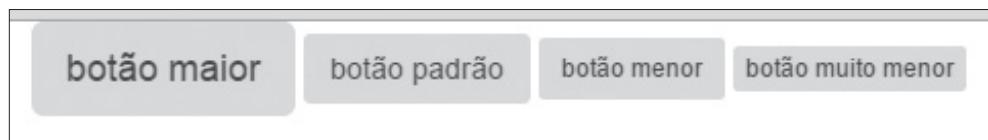
Figura 3.18 – Dimensões de botões.

Por padrão os botões são elementos do tipo `inline`. A classe `btn-block` faz com que os botões se comportem como elementos nível de bloco e se expandam em largura por toda a largura do elemento que os contêm.

Use a classe `active` para estilizar um botão tal como quando ele recebe o foco (cor de fundo mais escura).

Para desabilitar um botão use o atributo `disabled`. Botões desabilitados são estilizados com uma cor de fundo mais clara.

A marcação HTML para criar aplicar estas classes aos botões é conforme a seguir.

• **HTML**

```
<div class="container">
   <div class="row">
      <div class="col-md-6">
         <button type="button" class="btn btn-info btn-block">botão nível de bloco</button>
         <button type="button" class="btn btn-success btn-block">botão nível de bloco</button>
         <button type="button" class="btn btn-danger btn-block">botão nível de bloco</button>
      </div>
      <div class="col-md-6">
         <button type="button" class="btn btn-info active">botão estado ativo</button>
         <button type="button" class="btn btn-success active">botão estado ativo</button>
         <button type="button" class="btn btn-danger active">botão estado ativo</button>
      </div>
   </div> <!-- /.row-->
<hr>
   <div class="row">
      <div class="col-md-6">
         <button type="button" class="btn btn-info disabled">botão desabilitado</button>
         <button type="button" class="btn btn-success disabled">botão desabilitado</button>
         <button type="button" class="btn btn-dan</button>
      </div>
   </div> <!-- /.row -->
</div> <!-- /.container -->
```

[.../c3/outras-classes-para-botoes.html]

3.5.1.3 Estilização de checkbox

Para os controles `input` tipo `checkbox` está prevista a classe `checkbox` que serve de seletor para regras CSS destinadas basicamente a definir `padding`, `margin` e alinhamentos com o objetivo de obter uma renderização crossbrowser para esses controles.

A marcação HTML para cada checkbox e seu rótulo deverá estar contida dentro de um elemento `div` que recebe a classe `checkbox` conforme mostrado a seguir.

• **HTML**

```
<div class="checkbox">
   <label><input type="checkbox">Opção1</label>
</div>
<div class="checkbox">
```

```
    <label><input type="checkbox">Opção2</label>
</div>
<div class="checkbox">
    <label><input type="checkbox">Opção3</label>
</div>
```

Os controles são renderizados na vertical. Opcionalmente você poderá usar a classe `checkbox-inline` a ser declarada para o elemento `label`, que provoca a renderização na horizontal conforme mostrado a seguir.

- **HTML**

```
<label class="checkbox-inline"><input type="checkbox">Opção1</label>
<label class="checkbox-inline"><input type="checkbox">Opção2</label>
<label class="checkbox-inline"><input type="checkbox">Opção3</label>
```

É possivel estilizar os campos checkbox em formato de botões conforme mostrado na marcação HTML a seguir.

- **HTML**

```
<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary active">
        <input type="checkbox" autocomplete="off" checked>Opção 1 (marcado)
    </label>
    <label class="btn btn-primary">
        <input type="checkbox" autocomplete="off">Opção 2
    </labey">
        <input type="checkbox" autocomplete="off">Opção 3
    </label>
</div>
```

[.../c3/estilizacao-de-checkbox.html]

Na figura 3.19 mostramos a renderização dos controles na vertical e na horizontal e também a estilização dos checkboxes em formato de botão, conforme as marcações anteriores.
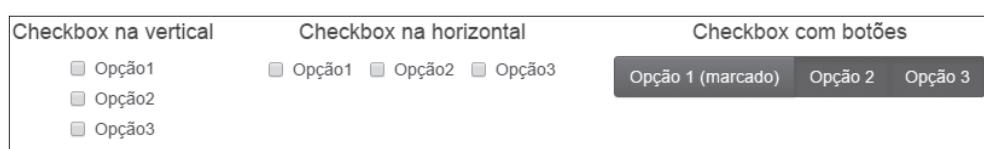
Figura 3.19 – Checkbox na vertical, horizontal e botões.

3.5.1.4 Estilização de botão radio

A marcação HTML para cada botão radio e seu rótulo deverá estar contida dentro de um elemento `div` que recebe a classe `radio` conforme mostrado a seguir.

- **HTML**

```
<div class="radio">
   <label><input type="radio" name="opcoes1">Opção1</label>
</div>
<div class="radio">
   <label><input type="radio" name="opcoes1">Opção2</label>
</div>
<div class="radio">
   <label><input type="radio" name="opcoes1">Opção3</label>
</div>
```

Os controles são renderizados na vertical. Opcionalmente você poderá usar a classe `radio-inline` a ser declarada para o elemento `label`, que provoca a renderização na horizontal conforme mostrado a seguir.

- **HTML**

```
<label class="radio-inline"><input type="radio" name="opcoes2">Opção1</label>
<label class="radio-inline"><input type="radio" name="opcoes2">Opção2</label>
<label class="radio-inline"><input type="radio" name="opcoes2">Opção3</label>
```

É possivel estilizar os campos tipo radio em formato de botões conforme mostrado na marcação HTML a seguir.

```
<div class="btn-group" data-toggle="buttons">
   <label class="btn btn-primary">
      <input type="radio" name="opcao" id="opcao1" autocomplete="off">Radio 1
   </label>
   <label class="btn btn-primary">
      <input type="radio" name="opcao" id="opcao2" autocomplete="off">Radio 2
   </label>
   <label class="btadio" name="opcao" id="opcao3" autocomplete="off">Radio 3
   </label>
</div>
```

[.../c3/estilização-de-botao-radio.html]

Na figura 3.20 mostramos a renderização dos controles na vertical e na horizontal e também a estilização dos campos tipo radio em formato de botão, conforme as marcações anteriores.
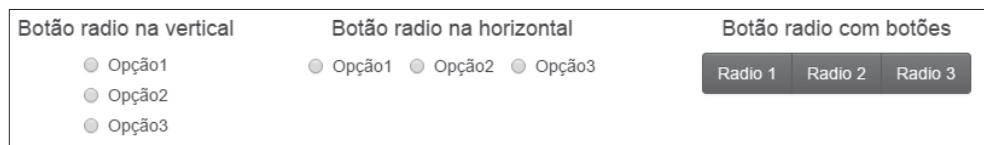
Figura 3.20 – Botão radio na vertical, na horizontal e botões.

3.5.1.5 Estilização de select

Os controles do tipo seleção devem ser marcados com a classe `form-control` para recebem uma estilização padrão que consiste simplesmente na aplicação de uma borda cinza com cantos arredondados. A largura deste elemento é igual a largura do seu container.

Alerta: Os navegadores Internet Explorer versões anteriores ao IE9 não suportam bordas arredondadas.

A estilização da borda é aplicada mesmo quando se declara o atributo `multiple` para o elemento `select`. A marcação HTML mínima para mostrar a renderização estilizada desses controles é conforme a seguir

- **HTML**

```
<select class="form-control">
   <option>Escolha uma das opções</option>
   <option>Opção1</option>
   <option>Opção 2</option>
   ...
</select>
 <select class="form-control" multiple>
/option>
   <option>Opção 1</option>
   <option>Opção 2</option>
   ...
</select>
```

[.../c3/estilizacao-de-select.html]

Na figura 3.21 mostramos a renderização dos controles com e sem a definição do atributo `multiple` conforme a marcação anterior.

Figura 3.21 – Controle de seleção.

3.5.1.6 Texto de ajuda em controles

O Boostrap prevê a classe `help-block` que se destina a posicionar os textos de ajuda para preenchimento ou mensagens de validação de controles.

O exemplo mostrado a seguir esclarece a definição desta classe para criar textos de ajuda em dois campos.

- **HTML**

```
<input type="text" class="form-control" placeholder="Seu CPF" aria-
describedby="helpBlock1">
   <span id="helpBlock1" class="help-block">somente números</span>
<input type="text" class="form-control" placeholder="y="helpBlock2">
   <span  id="helpBlock2" class="help-block">dd/mm/yyyy</span>
```

[.../c3/estilizacao-texto-de-ajuda.html]

3.5.2 Estilização de formulário

O Bootstrap prevê uma estilização padrão para formulários, bastando que se marque um `div` container com a classe `form-group` para cada conjunto de controles do formulário.

Está prevista a classe `form-inline` a ser declarada para o elemento `form` e cuja finalidade é estilizar o formulário em linha.

3.5.2.1 Estilização padrão e em linha

O código mostrado a seguir é a marcação HTML para um formulário simples e será usado para mostrar as estilizações padrão e em linha aplicada pela folha de estilos do Bootstrap.

- **HTML**

```
<form>
   <div class="form-group">
      <label for="email">Email</label>
      <input type="email" class="form-control" id="email" placeholder="Entre seu email">
   </div>
   <div class="form-group">
      <label for="senha">Senha</label>
      <input type="password" class="form-control" id="senha" placeholder="Senha">
   </div>
   <div class="form-group">
      <div class="checkbox">
      <label>Escolha as cores
         ...
    </div>
  <button type="submit" class="btn btn-
...
<form class="form-inline">
  <--! igual ao formulário anterior -->
</form>
```

[.../c3/estilizacao-padrao-e-em-linha-de-formulario.html]

Na figura 3.22 mostramos a renderização padrão e em linha para o formulário, conforme a marcação anterior.

Figura 3.22 – Formulário com estilização padrão e em linha.

3.6 Imagens

O Bootstrap prevê a aplicação de três efeitos simples em imagens inseridas com uso de marcação HTML. Os efeitos são aplicados quando se define os valores do atributo classe para o elemento `img` conforme listados a seguir.

- `img-rounded` – Aplica na imagem cantos arredondados usando raio igual a 6px.
- `img-circle` – Aplica na imagem cantos arredondados usando raio igual a 50%. A imagem é renderizada como um círculo ou uma elipse, dependendo do seu aspect-ratio.
- `img-thumbnail` – Aplica na imagem um `padding` de 4px e uma borda de 1px na cor cinza.

No Bootstrap, por padrão, as imagens não são responsivas, assim, para fazer a imagem responsiva use a classe `img-responsive`.

Se você pretende que todas as imagens da sua aplicação sejam responsivas, por padrão, crie a seguinte regra CSS.

- **HTML**

```
img {
  display: block;
  muto;
}
```

Para visualizar a estilização das imagens com os três efeitos aplicados consulte o arquivo disponível no site do livro conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<img class="img-responsive img-circle" src="imagem.jpg" alt="dinossauro">
<img class="img-responsive img-rounsauro">
<img class="img-responsive img-thumbnail" src="imagem.jpg" alt="dinossauro">
```

[.../c3/estilizacao-de-imagem.html]

Na figura 3.23 mostramos o efeito em imagens resultantes da definição dessas classes, conforme marcação HTML anterior.

Figura 3.23– Estilização de imagem.

3.7 Classes auxiliares

O Bootstrap prevê uma série de classes denominadas de *helper classes* que se destinam a destacar, centrar, flutuar, esconder, textos e componentes bem como destacar fundos, inserir marcas visuais fornecer dicas para leitores de tela e navegação por teclado. Veremos, a seguir, a aplicação e efeitos de tais classes.

3.7.1 Cores contextuais

Para estilizar textos com uma cor estão disponíveis as seguintes classes.

- `text-muted` – Estiliza textos na cor #`777`.
- `text-primary` – Estiliza textos na cor `#337ab7`.
- `text-success` – Estiliza textos na cor `#3c763d`.
- `text-info` – Estiliza textos na cor `#31708f`.
- `text-warning` – Estiliza textos tiliza textos na cor `#a94442`.

Para visualizar a estilização de textos com estas cores consulte o arquivodisponível no site do livro conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<p class="text-muted">Este é um texto estilizado com a classe <code>muted
   </code></p>
<p class="text-primary">Este é um texto estilizado com a classe <code>text-primary
   </code></p>
```

```
<p class="text-success">Este é um texto estilizado com a classe <code>text-success
   </code></p>
<p class="text-info">Este é um texto estilizado com a classe <code>text-info
   </code></p>
<p class="text-warning">Este é um texto estilizado com a classe <code>text-warning
   </code></p>
<p class="text-danger">Este é um texto estilizado com a classe <code>text-danger
   </code></p>
```

[.../c3/estilizacao-de-textos-com-cores.html]

3.7.2 Fundos contextuais

Para estilizar o fundo de um box com uma cor estão disponíveis as seguintes classes.

- `bg-primary` – Estiliza textos na cor `#337ab7`.
- `bg-success` – Estiliza textos na cor `#dff0d8`.
- `bg-info` – Estiliza textos na cor `#d9edf7`.
- `bg-warning` – Estiliza textos na cor `#fcf8e3`.
- `bg-danger` – Estiliza textos na cor `#f2dede`.

Para visualizar a estilização de fundo de boxes com estas cores consulte o arquivo disponível no site do livro conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<p class="bg-muted">Este é um box com cor de fundo estilizada com a classe <code>muted
   </code></p>
<p class="bg-primary">Este é um box com cor de fundo estilizada com a classe <code>
   bg-primary</code></p>
<p class="bg-success">Este é um box com cor de fundo estilizada com a classe
   <code>bg-success</code></p>
<p class="bg-info">Este é um box com cor de fundo estilizada com a classe <code>bg-info
   </code></p>
<p class="bg-warning">Este é um box com cor de fundo estilizada com a classe
   <code>bg-warning</code></p>
<p class="bg-danger">Este é um box com cor de fundo estilizada com a classe
   <code>bg-danger</code></p>
```

[.../c3/estilizacao-de-fundos-com-cores.html]

3.7.3 Marcadores fechar e dropdown

Para criar um marcador X (xis) que indica fechamento de uma janela usa-se a classe `close` em um elemento `button` combinado com um elemento `span` para marcar o xis. Para criar um marcador tipo pequeno triângulo apontando para baixo, destinado a indicar abertura de um e `caret` em um elemento `span`.

Para visualizar a estilização e criação destes dois marcadores consulte o arquivo disponível no site do livro conforme marcação HTML típica, mostrada a seguir. Notar que por padrão a marca para fechar é posicionada no canto superior à direita do container.

- **HTML**

```
<button type="button" class="close" aria-label="Close"><span aria-hidden="true">&times;
</span></button>
<span class="caret"></span>
```

[.../c3/estilizacao-de-marcadores.html]

3.7.4 Flutuar

Para flutuar elemento as classes `.pull-left` e `.pull-right` respectivamente.

Para visualizar o efeito destas classes consulte o arquivo conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<p class="pull-left">Parágrafo à esquerda</p>
<p class="pull-right">Parágrafo à direita</p>
```

[.../c3/estilizacao-com-float.html]

Alerta: Estas classes não devem ser usadas em barras de navegação. Conforme veremos no capítulo 4, para barras de navegação existem as classes `navbar-left` e `navbar-right`.

3.7.5 Clearfix a classe `clearfix`

Para "clarear floats" use a classe clearfix no container dos elementos flutuados.

Para visualizar o efeito destas classes consulte o arquivo conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<div class="clearfix">
   <p class="pull-left">Parágrafo à esquerda</p>
   <p class="pull-right">Parágrafo à direita</p>
</div>
```

[.../c3/estilizacao-com-clearfix.html]

3.7.6 Centrar na horizontal

Para centrar na horizontal use a classe center-block.

Para visualizar o efeito desta classe consulte o arquivo conforme marcação HTML típica, mostrada a seguir.

- **HTML**

```
<p class="center-block" style="width:50%">
   Lorem ipsum...
</div>
```

[.../c3/estilizacao-para-centrar.html]

3.7.7 Mostrar e esconder

Para revelar e esconder conteúdos, *inclusive para leitores de tela*, use as classes `.show` e `.hidden` respectivamente.

3.7.8 Leitores de tela e navegação por teclado

Para esconder conteúdos *exceto para leitores* de tela use a classe `sr-only`. Os conteúdos escondidos com esta classe devem ser revelados obrigatoriamente para usuários que navegam com uso do teclado. Para isso existe a classe `sr-only-focusable` deve ser usada em conjunto com `.sr-only`.

3.7.9 Image replacement

Esconder um texto dentro de um elemento e substituí-lo por imagem é uma técnica conhecida como image replacement. Nestes casos, para esconder o texto use a classe `text-hide`.

3.7.10 Mostrar e esconder por breakpoints

O Bootstrap prevê uma série de classes para controlar a visibilidade de conteúdos de acordo com o breakpoint. É possível mostrar conteúdos em larguras de viewport acima de determinada largura e escondê-los abaixo daquela largura e vice-versa.

Convém ressaltar que a opção pelo uso destas classes deve ser resultante de uma cuidadosa análise e conveniência, pois uma das diretrizes fundamentais do design responsivo preconiza que não se deve esconder conteúdos em função de larguras de telas ou natureza dos dispositivos.

Nas tabelas 3.1 e 3.2 mostramos a sintaxe destas classes e seus efeitos.

Tabela 3.1– Classes responsivas sintaxe

	Telefones (<768px)	Tablets(>=768px)	Desktop (>=992px)	Desktop (>=1200px)
.visible-xs-*	MOSTRA	ESCONDE	ESCONDE	ESCONDE
.visible-sm-*	ESCONDE	MOSTRA	ESCONDE	ESCONDE
.visible-md-*	ESCONDE	ESCONDE	MOSTRA	ESCONDE
.visible-lg-*	ESCONDE	ESCONDE	ESCONDE	MOSTRA
.hidden-xs	ESCONDE	MOSTRA	MOSTRA	MOSTRA
.hidden-sm	MOSTRA	ESCONDE	MOSTRA	MOSTRA
.hidden-md	MOSTRA	MOSTRA	ESCONDE	MOSTRA
.hidden-lg	MOSTRA	MOSTRA	MOSTRA	ESCONDE

Tabela 3.2– Classes responsivas níveis

Grupo de classes	CSS display
.visible-*-block	display: block;
.visible-*-inline	display: inline;
.visible-*-inline-block	display: inline-block;

Observe a seguir um exemplos de sintaxe para algumas destas classes.

- **HTML**

```
<span class="hidden-xs">Esconde em x-small</span>
<span class="visible-xs-block">Visivel em x-small</span>
...
<span class="hidden-sm">Esconde em small</span>
<span class="visible-sm-block">Visível em small</span>
...
<span class="hidden-md">Esconde em medium</span>
<span class="visible-md-block">Visível em medium</span>
...
<span class="hidden-lg">Esconde em large</span>
<span class="visible-lg-block">Visível em large</span>
```

[.../c3/estilizacao-mostra-esconde-por-breakpoints.html]

3.7.11 Classes para impressão

O Bootstrap prevê uma série de classes para controlar os conteúdos a serem impressos.

Na tabela 3.3 mostramos a sintaxe destas classes e seus efeitos.

Tabela 3.3– Classes para impressão

Classes	Navegador	Impressão
.visible-print-block	ESCONDE	MOSTRA
.visible-print-inline	ESCONDE	MOSTRA
.visible-print-inline-block	ESCONDE	MOSTRA
.hidden-print	MOSTRA	ESCONDE

3.8 Validação de formulários

A indicação do resultado da validação faz-se com uso de cores e adicionalmente com inserção de ícones.

3.8.1 Validação com cores

O Bootstrap prevê três classes para estilizar com cores os rótulos e as bordas dos campos do formulário fornecendo ao usuário informação sobre a validação dos dados entrados no campo.

Para cumprir critérios de acessibilidade forneça, para as tecnologias assistivas e para usuários impedidos de distinguir cores alternativas à cor conforme prescreve as especificações do W3C. Em campos não validados inclua a declaração `aria-invalid="true"`.

As classes para estilizar com cores e suas finalidades são listadas a seguir.

- `has-warning` – Estiliza as bordas, o texto do rótulo e textos de ajuda do controle na cor `#8a6d3b` e indica que a validação do campo resultou em um status de alerta.
- `has-error` – Estiliza as bordas, o texto do rótulo e textos de ajuda do controle na cor `#a94442`e indica que a validação do campo resultou em um status de erro.
- `has-success` – Estiliza as bordas, o texto do rótulo e textos de ajuda do controle na cor `#3c763d` e indica que a validação do campo resultou em um status de ação bem sucedida.

Essas classes podem ser definidas para os rótulos dos controle, para os próprios controles ou para os containers marcados com a classe `help-block`.

Você pode visualizar os efeitos da definição dessas classes consultando o arivro, mostrado a seguir.

- **HTML**

```
<div class="form-group has-success">
   <label class="control-label" for="sucesso">Input com sucesso</label>
   <input type="text" class="form-control" id="sucesso">
</div>
<div class="form-group has-warning">
   <label class="control-label" for="alerta">Input com alerta</label>
   <input type="text" class="form-control" id="alerta">
</div>
<div class="form-group has-error">
   <label class="control-label" for="erro">Input com erro</label>
   <input type="text" class="form-control" id="erro">
</div>
```

```
<div class="has-success">
   <div class="checkbox">
   <label>
      <input type="checkbox" id="checkboxSucesso" value="opcao1">
         Checkbox com sucesso
   </label>
   </div>
</div>
<div class="has-warning">
   <div class="checkbox">
   <label>
      <input type="checkbox" id="checkboxAlerta" value="opcao1">
      Checkbox com alerta
   </label>
   </div>
</div>
<div class="has-error">
   <div class="checkbox">
   <label>
      <input type="checkbox" id="checkboxErro" value="opcao1">
      Checkbox com erro
   </label>
   </div>
</div>
```

[.../c3/estilizacao-estado-de-validacao.html]

3.8.2 Validação com ícones

Adicionalmente ao uso de cores em campos validados podemos incluir um ícone dentro e à direita do campo validado. Neste caso use a classe `has-feedback` como mostrado no exemplo.

A marcação típica e as classes para incluir ícones de validação em campos de formulário é mostrada em destaque no exemplo a seguir.

- **HTML**

```
<div class="form-group has-sucess has-feedback">
   <label class="control-label" for="sucesso">Input com sucesso</label>
   <input type="text" class="form-control" id="sucesso">
```

```
    <span class="glyphicon glyphicon-ok form-control-feedback" aria-hidden="true"></span>
</div>
<div class="form-group has-warning has-feedback">
    <label class="control-label" for="alerta">Input com alerta</label>
    <input type="text" class="form-control" id="alerta">
    <span class="glyphicon glyphicon-warning-sign form-control-feedback" aria-
hidden="true"></span>
</div>
<div class="form-group has-error has-feedback">
    <label class="control-label" for="erro">Input com erro</label>
    <input type="text" class="form-control" id="erro">
    <span class="glyphicon glyphicon-remove form-control-feedback" aria-hidden="true"></
span>
</div>
```

[.../c3/ estilizacao-icones-de-validacao.html]

Dica: Mais detalhes sobre ícones do Bootstrap em [4.1]

3.9 Temas

Ainda que não seja do escopo deste livro estudar a criação de temas para o Bootstrap ao encerrar este capítulo vamos mencionar algumas facilidades que visam a criar temas para um projeto desenvolvido com uso do Bootstrap.

Conforme vimos no item [1.3] o download dos arquivos do Bootstrap nos fornece, além de outros, o arquivo *bootstrap.css* e também sua versão minificada *bootstrap.min.css*. Este arquivo contém todas as regras CSS para estilização dos inúmeros seletores e componentes e foi criado para atender um tema cuja apresentação é simples e ao mesmo tempo visualmente agradável.

O fato de o Bootstrap ser usado em larga escala no mundo todo faz com que uma aplicação ou site desenvolvido com uso do tema padrão seja imeditamente reconhecido e não raro comparado com outros sites que também usam aquele tema (eu diria que tornou-se um tema “batido” ou “manjado”).

Tal como acontece com qualquer folha de estilo destinada a criar um visual para o site (um tema) o desenvolvedor poderá alterar a folha de estilo de modo a criar um tema personalizado. Por tratar-se de uma folha de estilos com mais de 6.500 linhas em sua versão não minificada, a tarefa de alterá-la manualmente visando a criar um novo tema seria extremamente árdua.

Felizmente existem vários mecanismos que facilitam a tarefa conforme descritos a seguir.

- **Variáveis LESS** – Encontra-se disponível no site do Bootstrap uma interface gráfica destinada a alterar as váriáveis LESS que geram a folha de estilos. A interface encontra-se em *http://getbootstrap.com/customize/#less-variables*.
- **Interfaces** – Encontra-se disponível na internet várias interfaces gráficas para personalização e download de temas. Umas semelhantes à interface LESS do Bootstrap e outras de uso gratuito, mais intuitivo e simples. Entre estas indicamos a que se encontra em *http://bootstrap-live-customizer.com/*. Nela você escolhe o componente a estilizar e ao alterar uma das propriedades CSS visualiza imediatamente seu efeito. Uma busca no Google pela palavra--chave "Bootstrap themes generator" retorna inúmeros links que fornecem interfaces geradoras de temas.
- **Temas prontos** – Encontra-se disponível na internet vários temas prontos, sendo uns gratuitos e outros pagos. Este o caminho mais rápido para se encontrar um tema para o site pois na maioria dos casos é fornecida uma folha de estilos CSS minificada para download e basta que se substitua o link para a folha de estilo nativa por um link para a folha de estilo fornecida para o tema escohido. Ns do livrodos l mostram apontam para os arquivos dos exemplos aqui constantes criamos botões que permitem escolher direntes temas. Os temas ali constantes são gratuitos e foram obtidos em *https://bootswatch.com/*. Uma busca no Google pela palavra-chave "Bootstrap themes" retorna inúmeros links para sites que fornecem temas prontos para o Bootstrap.

CAPÍTULO 4

Componentes

Neste capítulo estudaremos os componentes de interface (ou widgets) do Bootstrap. Componentes são criados com blocos de código padrão e classes de estilização previstas pelo Bootstrap de modo a renderizar o componente de forma apropriada ao padrão de estilização, ou tema, adotado pelo Bootstrap. Para cada componente examinaremos a marcação HTML típica e as classes de estilização necessárias para renderização correta do componente. Os componentes disponíveis e que serão estudados neste capítulo são: ícones, dropdowns, agrupamento de botões, botões dropdown, elementos de navegação, rótulos e marcadores, thumbnails, elementos tipográficos, alertas, barras de progresso, objetos de mídia e classes genéricas.

4.1 Ícones

O Bootstrap incorpora um conjunto de mais de 260 ícones denominado Glyphicon Halflings (*http://glyphicons.com/*) desenvolvido por terceiros e cedido por seus criadores para o Bootstrap.

Trata-se de um conjunto de ícones do tipo icon font que poderá ser inserido com uso de qualquer elemento da marcação, que não se destine a criar uma funcionalidade do Bootstrap, isto é, o elemento não deverá conter qualquer par atributo/valor específico do Bootstrap. Em geral usa-se um elemento `span` para inserir ícone inline e `div` para inserir ícone nível de bloco.

O elemento que marca o ícone deverá estar vazio, ou seja, nenhum conteúdo deverá ser inserido no elemento.

São previstas duas classes para inserir ícones: `glyphicon` e `glyphicon-*`, a primeira contém as regras CSS de estilização geral, válidas para todos os ícones do Bootstrap

e a segunda define o ícone a ser inserido. A marcação típica para inserir um ícone é mostrada a seguir.

```
<span class="gliphicon glyphicon-heart"></span>
```

Esta marcação cria o ícone de um coração (heart) na cor preta e com dimensões conforme explicado a seguir.

Icon font é um glifo, tal como é qualquer fonte, portanto pode ser estilizado como fontes. Os ícones são na cor preta conforme a cor padrão do Bootstrap para fontes e suas dimensões (e cores) são as mesmas das fontes contidas no elemento que insere o ícone, ou seja, em tese, um ícone inserido com uso de um elemento `h1` é maior do que um ícone inserido com uso de um elemento `p`, e sua cor a mesma cor daqueles elementos. Mas, essa comparação foi apenas para exemplificação, não insira ícones com uso desses elementos, pois eles não são cabeçalhos e muito menos parágrafos.

Observe um exemplo real que comprova a comparação feita.

```
<h1>Coração <span class="glyphicon glyphicon-heart"></span></h1>
<p>Coração <span class="glyphicon glyphicon-heart"></span></p>
```

[.../c4/icones.html]

Este exemplo encontra-se no site do livro, para consulta online.

No primeiro caso o tamanho de fonte do elemento `span` é herdado do elemento `h1` e no segundo do mesmo acontece com o elemento `p`.

Para alterar a cor do ícone use a propriedade CSS `color` que altera cor das fontes. Observe a seguir alguns exemplos de alteração de cores de ícones.

- **CSS**

```
.glyphicon { color: red; } /* Altera globalmente a cor de todos os ícones */
.glyphicon-heart { color: red; } /* Altera globalmente a cor de um determinado ícone */
```

Podemos usar uma das classes de cores contextuais conforme descritas em [3.7.1] para alterar a cor de um ícone, conforme mostrado a seguir.

```
<p>Coração <span class="glyphicon glyphicon-heart text-info"></span></p>
```

Podemos usar uma classe personalizada para alterar a cor de um ícone. A classe será estilizada com uso de CSS na cor escolhida pelo autor.

```
<p>Coração <span class="glyphicon glyphicon-heart cor-um"></span></p>
```

Tal como acontece com a cor de qualquer fonte, se não for explicitamente definida uma cor para o elemento que insere o ícone, ele ícone, herda a cor do elemento-pai.

Se por um motivo qualquer você precisar alterar as dimensões do ícone use a propriedade CSS `font-size` que altera tamanho de fontes e proceda tal como fizemos para alterar a cor do ícone. Para as dimensões do ícone, a herança CSS também é válida.

Para uma lista completa dos ícones e suas classes para inserção consulte *http://getbootstrap.com/components/#glyphicons*.

4.2 Dropdown e dropup

Convém ressaltar que para possibilitar o funcionamento do menu dropdown é necessário incluir na página que receberá o menu a biblioteca jQuery, o plugin do Bootstrap para dropdown (está no arquivo *bootstrap.min.js*) conforme mostrado em [1.4].

A marcação HTML típica, a definição das classes `dropdown` e `dropup` e dos atributos dos elementos de marcação necessários para estilizar e fazer funcionar um menu dropdown (abre submenu para baixo) e outro dropup (abre submenu para cima) são mostrados a seguir.

- **HTML**

```
1.  <div class="dropdown"> <!-- Menu abre para baixo -->
2.  <button class="btn btn-primary dropdown-toggle"  type="button" id="dropDown1"
    data-toggle="dropdown" aria-expanded="true">Frutas <span class="caret"></span></button>
3.  <ul class="dropdown-menu" role="menu" aria-labelledby="dropDown1">
4.      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Abacaxi</a></li>
5.      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Laranja</a></li>
6.      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Maçã</a></li>
7.      <li class="divider"></li>
8.      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Outras...</a></li>
9.  </ul>
10. </div> <!-- /.dropdown -->
11. <div class="dropup">  <!-- Menu abre para cima -->
12. <button class="btn btn-primary dropdown-toggle"  type="button" id="dropUp1"
    data-toggle="dropdown" aria-expanded="true">Frutas <span class="caret"></span></button>
13. <!-- idêntico ao anterior -->
14. </div> <!-- /.dropup -->
```

[.../c4/dropdown-dropup.html]

Código comentado:

Linha(s)	Descrição
Linhas 1 e 10	Para obter o efeito dropdown é necessário criar um `div` container e a ele definir `class=dropdown`. Esse `div` será o container geral do dropdown.
Linha 2	Aqui inserimos o elemento que ao ser clicado fará o menu abrir (no caso um botão). Esse elemento deverá receber o atributo `data-toggle="dropdown"`, pois atributos `data-*` são a referência para que o plugin dropdown que foi incorporado à página juntamente com a biblioteca jQuery identifique e faça funcionar o dropdown.
Linhas 3 e 9	Elemento `ul` container dos links do menu dropdown. Esse container deverá receber o atributo `class="dropdown-menu"`.
Linhas 4 a 8	Elementos `li` (itens da lista) containeres dos links do menu.
Linha 7	Aqui um elemento `li` (item da lista) vazio com a atributo `class="divider"`. Essa classe quando aplicada a um dos itens da lista que esteja vazio, cria uma linha horizontal cinza divisória dos itens.
Linhas 11 a 14	Cria um menu dropup de modo idêntico ao anterior. Esse menu abre para cima.
Linha 12	Marcação do botão para acionar o menu dropdown. As classes aqui contidas foram estudadas na linha 2.

Na figura 4.1 mostramos a renderização dos dois menus, conforme marcação HTML anterior.

Figura 4.1 – Menus dropdown e dropup.

No nosso exemplo usamos um elemento `button`, teoricamente você pode usar qualquer elemento HTML para acionar a abertura do menu, mas é claro, deverá escolher o elemento mais apropriado e semântico.

Dica: Alguns componentes estudados neste capítulo dependem da inclusão de plugins para funcionar. Nesses casos o Bootstrap prevê um atributo do tipo `data-*` com um valor declarado, que chama automaticamente o método do plugin no elemento ao qual foi aplicado. Como regra geral, sempre que você encontrar um atributo `data-*` na marcação, ele tem a função de chamar um plugin. Em geral o valor desse atributo é o nome do plugin (ver linha 2 do código comentado anteriormente). No capítulo 5 estudaremos, com detalhes, os plugins.

4.2.1 Alinhamento, título e item desabilitado

Por padrão, os componentes do Bootstrap são alinhados à esquerda segundo a convenção de escrita para idiomas ocidentais. Está previsto um valor para o atributo classe, denominado `dropdown-menu-right`, a ser declarado para o container dos itens do menu e que se destina a alinhar os itens do menu à direita com aplicação da declaração CSS `float: right;`.

Para inserir um título em um conjunto de itens do menu use a classe `dropdow-header` conforme marcação mostrada a seguir.

```
<li role="presentation" class="dropdown-header">Título dos itens</li>
```

Para desabilitar um item do menu use a classe `disabled` conforme marcação mostrada a seguir.

```
<li role="presentation" class="disabled">Item desabilitado</li>
```

O exemplo a seguir mostra a mesma marcação HTML do exemplo anterior para o menu dropup com a definição das classes para alinhar os itens do menu à direita, para criar um título para um conjunto de itens e para desabilitar um item.

- **HTML**

```
<div class="dropup">
<button class="btn btn-primary dropdown-toggle" type="button" id="dropDown1"
   data-toggle="dropdown" aria-expanded="true">Frutas <span class="caret"></span></button>
   <ul class="dropdown-menu dropdown-menu-right" role="menu" aria-labelledby="dropDown1">
      <li role="presentation" class="dropdown-header">Frutas específicas</li>
      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Abacaxi</a></li>
      <li role="presentation" class="disabled"><a role="menu-item" href="#"
         tabindex="-1">Laranja</a></li>
      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Maçã</a></li>
      <li class="divider"></li>
```

```
        <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Outras...</a></li>
    </ul>
</div> <!-- /.dropdown -->
```

[.../c4/dropdown-alinhamento-titulo-item-desabilitado.html]

4.3 Agrupamento de botões

O Bootstrap prevê as classes `btn-group`, `btn-group-vertical` e `btn-toolbar` destinadas a criar agrupamentos de botões. A seguir veremos cada uma dessas classes, sua aplicação e finalidade.

4.3.1 Agrupamento em linha

Para agrupar botões em linha basta criar um `div` container para os botões e a ele definir a classe `btn-group` e o atributo `role` conforme mostrado a seguir.

- **HTML**

```
<div class="btn-group" role="group">
    <button type="button" class="btn btn-primary">Novo</button>
    <button type="button" class="btn btn-primary">Abrir</button>
    <button type="button" class="btn btn-primary">Fechar</button>
    <button type="button" class="btn btn-primary">Salva
    <button type="button" class="btn btn-primary">Salvar</button>
</div>
```

[.../c4/agrupamento-de-botoes.html]

4.3.2 Agrupamento vertical

Para agrupar botões na vertical basta criar um `div` container para os botões e a ele definir a classe `btn-btn-group-vertical` conforme mostrado a seguir.

- **HTML**

```
<div class="btn-group-vertical" role="group">
    <button type="button" class="btn btn-primary">Novo</button>
    <button type="button" class="btn btn-primary">Abrir</button>
    <button type="button" class="btn btn-primary">Fechar</button>
```

```
    <button type="button" class="btn btn-primary">Salvar</button>
</div>
```

[.../c4/agrupamento-de-botoes.html]

4.3.3 Agrupamento barra de ferramentas

Para criar múltiplos grupos de botões basta criar um `div` container para todos os agrupamentos de botões (como estudados anteriormente) e a ele definir a classe `btn-toolbar` e `role="toolbar"` conforme mostrado a seguir.

- **HTML**

```
<div class="btn-toolbar" role="toolbar">
    <button type="button" class="btn btn-primary">Novo</button>
    <button type="button" class="btn btn-primary">Abrir</button>
    <button type="button" class="btn btn-primary">Fechar</button>
    <button type="button" class="btn btn-primary">Salvar</button>
</div>
```

[.../c4/agrupamento-de-botoes.html]

Na figura 4.2 mostramos a renderização dos três casos de agrupamento de botões, conforme marcações HTML anteriores.

4.3.4 Agrupamento justificado

Para agrupar botões em linha de modo que eles se expandam com igual largura e ocupem toda a largura do container basta criar um `div` container para cada um dos botões e um `div` container geral definindo para ele as classes `btn-group` e `btn-group-justified` e o atributo `role` conforme mostrado a seguir.

- **HTML**

```
<div class="btn-group btn-group-justified" role="group">
    <div class="btn-group role="group">
        <button type="button" class="btn btn-primary">Novo</button>
    </div>
    <div class="btn-group role="group">
        <button type="button" class="btn btn-primary">Abrir</button>
    </div>
```

```
    <div class="btn-group role="group">
        <button type="button" class="btn btn-primary">Fechar</button>
    </div>
    <div class="btn-group role="group">
        <button type="button" class="btn btn-primary">Salvar</button>
    </div>
</div>
```

[.../c4/agrupamento-de-botoes.html]

4.3.5 Dimensionamento

Conforme estudamos no item [3.5.1.2] por padrão, os botões são estilizados com um `padding` que determina suas dimensões. Em um agrupamento de botões as classes `.btn-group-lg`, `.btn-group-sm` e `.btn-group-xs`, possibilitam que se altere aquele valor padrão de `padding` e se obtenha botões com dimensões maiores, menores e muito menores que o botão padrão respectivamente.

A marcação HTML para criar tais botões é conforme a seguir.

- **HTML**

```
<div class="btn-group btn-group-lg" role="group" aria-label="Large button group">
    <button type="button" class="btn btn-success">novo</button>
    <button type="button" class="btn btn-success">abrir</button>
    <button type="button" class="btn btn-success">fechar</button>
    <button type="button" class="btn btn-success">salvar</button>
</div>
<div class="btn-group" role="group" aria-label="Default button group">
<!-- idem -->
</div>
<div class="btn-group btn-group-sm" role="group" aria-label="Medium button group">
<!-- idem -->
</div>
<<div class="btn-group btn-group-xs" role="group" aria-label="Small button group">
<!-- idem -->
</div>
```

[.../c4/agrupamento-de-botoes.html]

Observe na figura 4.2 os diferentes tipos de agrupamento de botões e seu dimensionamento, conforme as marcações mostradas no item [4.3].
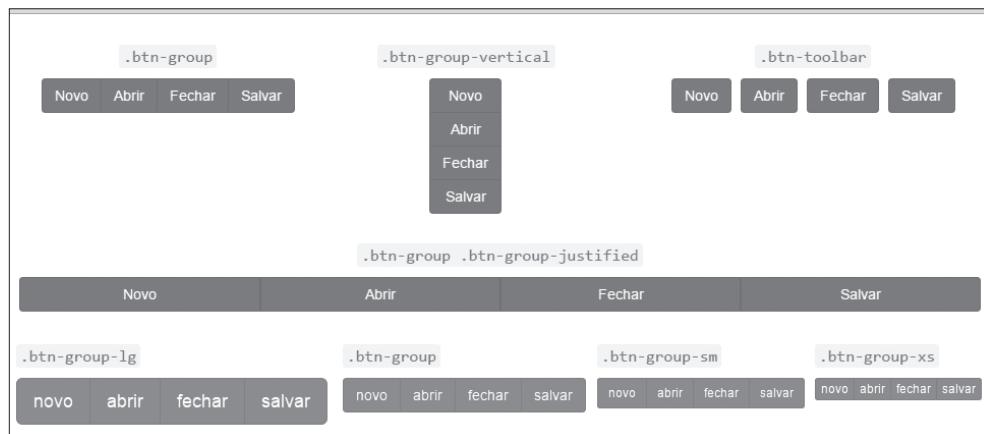
Figura 4.2 – Agrupamento de botões.

4.4 Botões dropdown

Convém ressaltar que para possibilitar o funcionamento de um botão dropdown é necessário incluir na página que receberá o botão a biblioteca jQuery, o plugin do Bootstrap para dropdown (está no arquivo *bootstrap.min.js*) conforme mostrado em [1.4].

Vejamos a seguir os casos possíveis de combinações para criar menus dropdown com botões.

4.4.1 Botões dropdown simples

Para criar um botão dropdown simples usamos um container geral com a classe `btn-group`. O botão é criado com qualquer uma das marcações para criar botões como mostrado em [3.5.1.2] e a ele deve ser definida a classe `dropdown-toggle` e o par atributo/valor `data-toggle="dropdown"`. A lista de links deve ser marcada com um elemento `ul` que recebe a classe `dropdown-menu`.

A marcação HTML típica para criar um botão dropdown, conforme descrito, é mostrada a seguir.

- **HTML**

```
<div class="btn-group">
   <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown"
aria-expanded="false">
      EDITAR <span class="caret"></span>
   </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">novo</a></li>
         <li><a href="#">abrir</a></li>
         <li><a href="#">fechar</a></li>
         <li class="divider"></li>
         <li><a href="#">salvar</a></li>
      </ul>
</div>
```

[.../c4/botoes-dropdown.html]

4.4.2 Botões com ícone de abertura separado

Para criar um botão dropdown simples como ícone de abertura separado usa-se a mesma marcação HTML anterior com acréscimo de um novo botão contendo o seu texto, conforme destacado no código a seguir.

- **HTML**

```
<div class="btn-group">
   <button type="button" class="btn btn-default>EDITAR</button>
   <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown"
      aria-expanded="false">
      EDITAR <span class="caret"></span>
   </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">novo</a></li>
         <li><a href="#">abrir</a></li>
         <li><a href="#">fechar</a></li>
         <li class="divider"></li>
         <li><a href="#">salvar</a></li>
      </ul>
</div>
```

[.../c4/botoes-dropdown.html]

4.4.3 Botões com ícone de abertura Glyphicon e integrado

A marcação HTML típica para criar o ícones glyphicon integrados aos botões consiste em se substituir a marcação do ícone de abertura padrão pela marcação do ícone glyphicon conforme mostrado em destaque a seguir.

- **HTML**

```
<div class="btn-group">
   <button type="button" class="btn btn-default>EDITAR</button>
   <button type="button" class="btn btn-default dropdown-toggle"
         data-toggle="dropdown" aria-expanded="false">
      <span class="caret"></span>
      <span class="glyphicon glyphicon-pencil"></span>
   </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">novo</a></li>
         <li><a href="#">abrir</a></li>
         <li><a href="#">fechar</a></li>
         <li class="divider"></li>
         <li><a href="#">salvar</a></li>
      </ul>
</div>
```

[.../c4/botoes-dropdown.html]

4.4.4 Dimensionamento

Conforme estudamos no item [3.5.1.2] por padrão, os botões são estilizados com um `padding` que determina suas dimensões. O uso das classes `.btn-group-lg`, `.btn-group-sm` e `.btn-group-xs`, possibilita que se altere aquele valor padrão de `padding` e se obtenha botões com dimensões maiores, menores e muito menores que o botão padrão respectivamente. Ver [4.3.5] para exemplo de uso destas classes.

4.4.5 Botão dropup

É possível inverter o sentido de abertura do botão dropdown com uso da classe `dropup` no container geral do botão como mostrado a seguir.

- **HTML**

```
<div class="btn-group dropup">
   <!-- marcação típica como mostrada nos exemplos anteriores -->
</div>
```

Observe na figura 4.3 os diferentes tipos add-on de texto e botões dropdown, conforme as marcações mostradas no item [4.4].

Figura 4.3 – Botões dropdown.

4.5 Controle input com add-on

O Bootstrap prevê funcionalidades que permitem inserir texto de ajuda (ou complementar) e botões, antes, depois ou mesmo em ambas as extremidades (antes e depois) dos controles (campos) de formulário `input` para entrada de textos.

Convém ressaltar que para possibilitar a inserção é necessário incluir na página que receberá o controle a biblioteca jQuery, o plugin do Bootstrap para dropdown (está no arquivo *bootstrap.min.js*) conforme mostrado em [1.4].

Vejamos a seguir os casos possíveis de combinações para inserção em controles `input`.

4.5.1 Texto add-on

Para obter esses efeitos o Bootstrap prevê a classe `input-group` a ser declarada em um elemento `div` container para o texto e o campo `input`. É prevista também a classe `input-group-addon` a ser declarada em um elemento `span` container para o texto.

A marcação HTML para mostrar a renderização estilizada para esses casos é conforme a seguir.

- **HTML**

```
<!-- Texto integrado antes -->
<div class="input-group">
   <span class="input-group-addon" id="addon1">@</span>
   <input type="text" class="form-control" placeholder="Twitter"
      aria-describedby="addon1">
</div>
<!-- Texto integrado depois -->
<div class="input-group">
   <input type="text" class="form-control" placeholder="Seu email"
      aria-describedby="addon2">
   <span class="input-group-addon" id="addon2">@gmail.com</span>
</div>
<!-- Texto integrado antes e depois -->
<div class="input-group">
   <span class="input-group-addon">R$</span>
   <input type="text" class="form-control" placeholder="Total da sua compra"
      aria-label="Valor em reais">
   <span class="input-group-addon">,00</span>
</div>
```

[.../c4/input-com-add-on.html]

Notar que ordem de declaração dos elementos `span` e `input` é importante para a correta renderização do texto na sua posição (antes, depois ou em ambos os lados do controle).

4.5.2 Botão simples add-on

Assim como vimos para textos no item anterior, os controles `input` para textos podem ser estilizados com uso de botões integrado antes ou depois do campo `input`.

Para obter esses efeitos o Bootstrap prevê a classe `input-group-btn` a ser declarada no elemento `span` container para o botão.

A marcação HTML para mostrar a inserção de botão simples antes e depois do controle é conforme a seguir.

- **HTML**

```
<!-- Botão simples antes -->
<div class="input-group">
   <span class="input-group-btn">
      <button class="btn btn-default" type="button">IR</button>
   </span>
   <input type="text" class="form-control" placeholder="CPF">
</div>
<!-- Botão simples depois -->
<div class="input-group">
   <input type="text" class="form-control" placeholder="Palavra-chave">
   <span class="input-group-btn">
      <button class="btn btn-default" type="button">BUSCAR</button>
   </span>
</div>
```

[.../c4/input-com-add-on.html]

Notar que ordem de declaração dos elementos que marcam o botão e o campo `input` é importante para a correta renderização do botão na sua posição.

4.5.3 Botão dropdown add-on

Assim como vimos para botões simples no item anterior, os controles `input` para textos podem ser estilizados com uso de botões dropdown integrado antes ou depois do campo `input`. O botão dropdown deve estar contido em um `div` com a classe `input-group-btn`.

Para obter esses efeitos basta que se declare a marcação HTML típica para botões dropdown em lugar de botões simples.

A marcação HTML para mostrar a inserção de botão dropdown antes e depois do controle é conforme a seguir.

- **HTML**

```
<!-- Botão dropdown antes -->
<div class="input-group">
   <div class="input-group-btn">
      <button type="button" class="btn btn-default">CONTATOS</button>
      <button type="button" class="btn btn-default dropdown-toggle"
```

```
            data-toggle="dropdown" aria-expanded="false">
         <span class="glyphicon glyphicon-envelope"></span>
         </button>
      <ul class="dropdown-menu" role="menu">
         <li><a href="#">um</a></li>
         <li><a href="#">dois</a></li>
         <li><a href="#">três</a></li>
         <li class="divider"></li>
         <li><a href="#">quatro</a></li>
      </ul>
      </div>
      <input type="text" class="form-control" placeholder="Contatos">
   </div>
   <!-- Botão dropdown depois -->
   <div class="input-group">
      <input type="text" class="form-control" placeholder="Favoritos">
      <div class="input-group-btn">
      <button type="button" class="btn btn-default">FAVORITOS</button>
         <!-- idênti
```

[.../c4/input-com-add-on.html]

Notar que ordem de declaração dos elementos que marcam o botão e o campo `input` é importante para a correta renderização do botão na sua posição.

Observe na figura 4.4 os diferentes tipos de add-on em controles `input`, conforme as marcações mostradas no item [4.5].
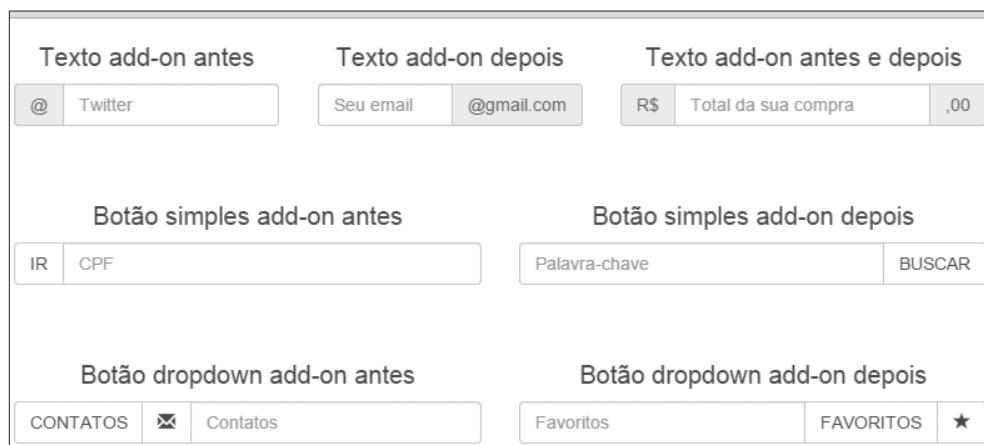
Figura 4.4 – Input com add-on.

4.6 Navegação

As funcionalidades de navegação previstas pelo Bootstrap são: abas e pills (lista de links simples). A marcação padrão para criar estes dois tipos de navegação consiste de um container `ul` com a classe `nav` e seus elementos `li`. As classes `nav-tabs` e `nav-pills` criam, respectivamente, os dois tipos de navegação.

Observe a marcação típica como citada e que será usada nos exemplos de navegação constantes deste item do livro.

- **HTML**

```
<ul class="nav">
   <li role="presentation" class="active"><a href="#">Home</a></li>
   <li role="presentation"><a href="#">Portfólio</a></li>
   <li role="presentation"><a href="#">Artigos</a></li>
   <li role="presentation"><a href="#">Contato</a></li>
</ul>
```

4.6.1 Aba básica

Para criar um mecanismo de navegação do tipo abas basta acrescentar a classe `nav-tabs` ao container `ul` da marcação padrão mostrada anteriormente.

- **HTML**

```
<ul class="nav nav-tabs">
   <li role="presentation" class="active"><a href="#">Home</a></li>
   <li role="presentation"><a href="#">Portfólio</a></li>
   <li role="presentation"><a href="#">Artigos</a></li>
   <li role="presentati
</ul>
```

[.../c4/navegacao.html]

4.6.2 Aba desabilitada

Para desabilitar uma aba defina a classe `disabled` para elemento `li` que marca o link.

- **HTML**

```
<ul class="nav nav-tabs">
   <li role="presentation" class="active"><a href="#">Home</a></li>
   <li role="presentation"><a href="#">Portfólio</a></li>
   <li role="presentation" class="disabled"><a href="#">Artigos</a></li>
   <li role="presentat>
</ul>
```

[.../c4/navegacao.html]

4.6.3 Aba com link dropdown

Para criar uma aba com link dropdown defina a classe `dropdown` para elemento `li` que marca a aba e crie a marcação padrão para link dropdown dentro daquele elemento `li` conforme mostrado a seguir.

- **HTML**

```
<ul class="nav nav-tabs">
   <li role="presentation" class="active"><a href="#">Home</a></li>
   <li role="presentation"><a href="#">Portfólio</a></li>
   <li role="presentation" class="dropdown">
      <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button"
         aria-expanded="false">
      Artigos <span class="caret"></span></a>
         <ul class="dropdown-menu" role="menu">
            <li><a href="#">HTML</a></li>
            <li><a href="#">CSS</a></li>
            <li><a href="#">javascript</a></li>
         </ul>
   </li>
   <li role="presentli>
</ul>
```

[.../c4/navegacao.html]

4.6.4 Abas justificadas

Para justificar as abas defina a classe `nav-justified` para elemento `ul` container das abas. Justificar tem o efeito de fazer com que as abas tenham a mesma largura e se estendam por toda a largura do container no qual foram inseridas.

- **HTML**

```
<ul class="nav nav-tabs nav-justified">
   <li role="presentation" class="active"><a href="#">Home</a></li>
   <li role="presentation"><a href="#">Portfólio</a></li>
   <li role="presentation"><a href="#">Artigos</a></li>
   <li role="prese</li>
</ul>
```

[.../c4/navegacao.html]

4.6.5 Pills

A marcação para criar a navegação tipo pills é idêntica à mostrada para criar tabs. Basta substituir a classe `nav-tabs` pela classe `nav-pills`. O exemplo mostrando todos os casos de navegação estudados nos itens anteriores encontra-se online.

Observe na figura 4.5 os diferentes tipos de navegação, conforme as marcações mostradas no item [4.6].
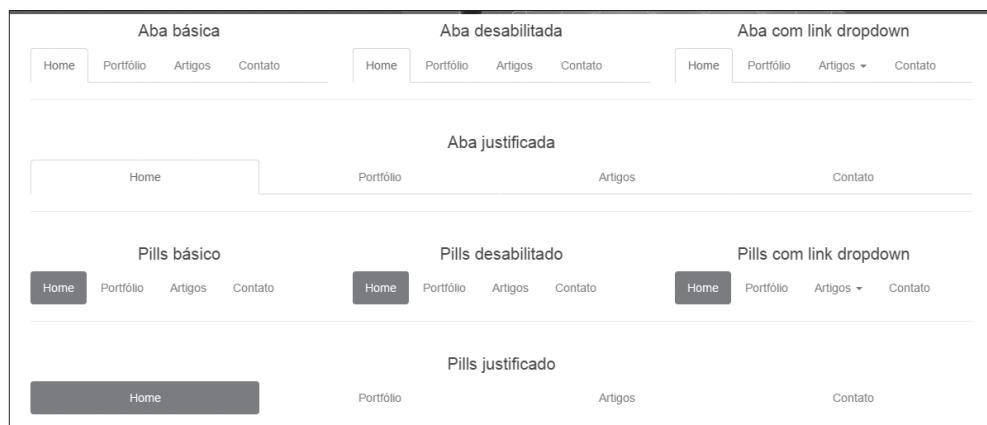
Figura 4.5 – Navegação tipos abas e pills.

4.7 Barra de navegação

O Bootstrap prevê funcionalidades que permitem criar uma grande variedade de barras de navegação contendo não somente links, mas imagens, textos, caixas de busca, elementos de formulário, dropdown etc. As barras podem ainda ser fixas no topo ou rodapé da aplicação ou rolar com o conteúdo. Neste item veremos todas as possibilidades de criação de barras de navegação.

As barras de navegação são responsivas, como são todos os componentes do Bootstrap. Barras de navegação são dispostas na horizontal em larguras de viewport acima de 768px. Abaixo desta largura as barras de navegação assumem uma disposição vertical típica da navegação móvel com um ícone tipo hamburger para revelar os links. Ao visualizar os exemplos redimensione a janela do navegador e observe o comportamento das diferentes barras de navegação mostradas.

O container geral de barras de navegação deverá ser o elemento `nav`, contudo se por qualquer motivo for necessário usar-se outro elemento, tal como um `div` deverá ser, para ele container, definido o par atributo/valor `role="navigation"` servido para tecnologias assistivas.

Convém ressaltar que para possibilitar o funcionamento das barras de navegação é necessário incluir na página que receberá o botão a biblioteca jQuery, o plugin do Bootstrap para dropdown (está no arquivo *bootstrap.min.js*) conforme mostrado em [1.4].

4.7.1 Barra de navegação básica

Trata-se de uma barra de navegação contendo um título (em geral o nome ou o logotipo da aplicação) e links.

Observe a marcação típica para criar uma barra de navegação básica.

- **HTML**

```
1. <nav class="navbar navbar-default">
2. <div class="container-fluid">
3. <div class="navbar-header">
4.     <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
             data-target="#barraBasica">
5.         <span class="sr-only">Toggle navigation</span>
6.         <span class="icon-bar"></span>
7.         <span class="icon-bar"></span>
```

```
8.          <span class="icon-bar"></span>
9.      </button>
10.     <a class="navbar-brand" href="#">Site</a>
11. </div>
12. <div class="collapse navbar-collapse" id="barraBasica">
13.     <ul class="nav navbar-nav">
14.         <li class="active"><a href="#">Home <span class="sr-only">(current)</span></a></li>
15.         <li><a href="#">Portfólio</a></li>
16.         <li><a href="#">Matérias</a></li>
17.         <li><a href="#">Contato</a></li>
18.     </ul>
19. </div><!-- /.navbar-collapse -->
 -->
21. </nav>
```

[.../c4/barra-de- navegacao.html]

Código comentado:

Linha(s)	Descrição
Linhas 1 e 21	Elemento `nav` container da barra de navegação. Definir para este container as classes `navbar` e `navbar-default`. Por padrão o fundo da barra de navegação é em cor clara. Se definirmos a classe `navbar-inverse` em lugar da classe `navbar-default` o fundo da barra será em cor escura.
Linha 2	Logo após o elemento `nav` deve ser marcado um elemento `div` com classe `container` ou `container-fluid`. A escolha de uma ou outra classe determina a largura total da barra. Se quisermos que a barra ocupe toda a largura da viewport, em qualquer situação, o elemento `nav` deve ser colocado como elemento-filho de `body` e não dentro de `div.row` e neste caso usa-se a classe `container-fluid` caso contrár`ontainer`. No exemplo disponível online no site do livro mostramos as duas situações.
Linhas 3 e 11	Container para o título do site e para o ícone hamburger que aparece em dispositivos móveis.
Linhas 4 a 9	Botão do ícone hamburger.
Linha 5	Dica para tecnologias assistivas. Notar a classe `sr-only` mostrar este conteúdo somente para aquelas tecnologias. Ver [1.7].
Linhas 6 a 8	Classe `icon-bar` para criar as três linhas do ícone hamburger.
Linha 10	Título do site. O destino deste link deve ser a home-page do site. Se em lugar do título do site você decidir por colocar uma pequena imagem (20px x 20px) do logotipo ou marca do site substitua o texto do link por um elemento `img` com o endereço do logotipo.

Linha(s)	Descrição (cont.)
Linhas 12 a 19	Container para o elemento `ul` que marca os links da barra de navegação. Notar que o valor do atributo `id` deste container é o mesmo do atributo `data-target` do botão na linha 4.
Linhas 14	Marcar dinamicamente o elemento `li` container do link para a página atual com a classe `active` estiliza o link diferenciadamente indicando visualmente que aquele link é o da página atual. Neste caso use a classe `sr-only` (ver [1.7]) para fornecer a mesma indicação às tecnologias assistivas. Marca os links da barra de navegação. Notar que o valor do atributo `id` deste.

O exemplo prático mostrando a construção de barras de navegação básicas encontra-se disponível, no site do livro, para consulta online. Ao visualizar o exemplo redimensione a janela do navegador para observar o comportamento da barra em dispositivos com pequenas larguras de viewport (abaixo de 768px).

Observe na figura 4.6 os diferentes tipos de barra de navegação, conforme as marcações mostradas no item [4.7.1].
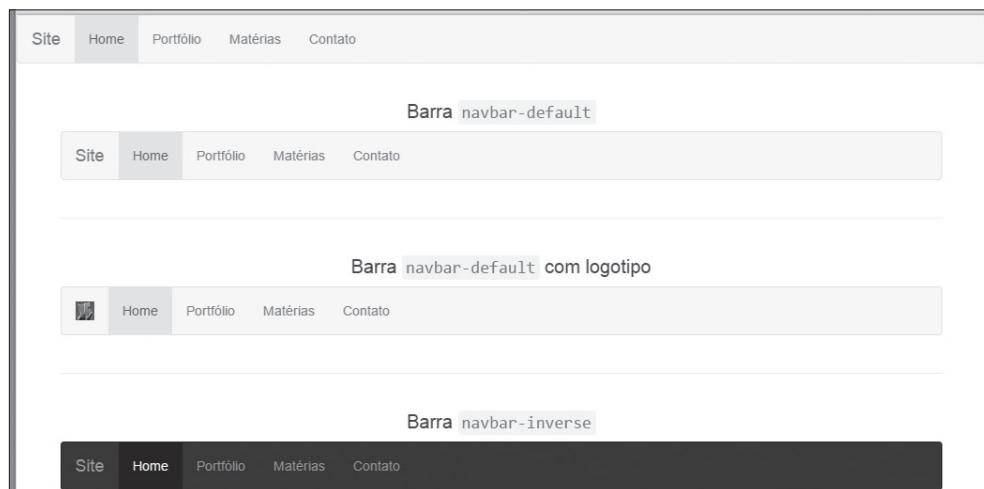
Figura 4.6 – Barra de navegação básica.

4.7.2 Barra de navegação fixa

Por padrão a barra de navegação rola acompanhando a rolagem da tela. As classes `navbar-fixed-top` e `navbar-fixed-bottom` quando declaradas para o elemento `nav` que marca a barra faz com que ela, barra permaneça fixa na borda superior ou na borda inferior da tela, ou seja, não rola com a tela.

Observe a marcação típica para criar uma barra de navegação fixa.

- **HTML**

```
<nav class="navbar navbar-default navbar-fixed-top"> /* Fixa na borda superior da tela */
...
```

ou

```
<nav class="navbar navbar-default navbar-fixed-button"> /* fixa na borda inferior da tela */
...
```

[.../c4/barra-de-navegacao-fixa.html]

O exemplo prático mostrando a construção de barras de navegação fixas encontra-se disponível, no site do livro, para consulta online. Ao visualizar o exemplo redimensione a janela do navegador para observar o comportamento da barra em dispositivos com pequenas larguras de viewport (abaixo de 768px).

Observe na figura 4.7 as barras de navegação fixas, conforme as marcações mostradas no item [4.7.2].

Figura 4.7 – Barra de navegação fixa.

Alerta: A altura padrão de uma barra de navegação é de 50px. Assim, quando se prevê barra de navegação fixa na página é necessário que se declare explicitamente para o elemento `body` da página um `padding-top` de 50px.

4.7.3 Barra de navegação com caixa de busca

É prática comum e amplamente utilizada inserir uma caixa de busca na barra de navegação. O Bootstrap prevê a classe `navbar-form` a ser declarada no elemento `form`, container da caixa de busca, para essa inserção. Adicionalmente use as classes gerais `navbar-left` e `navbar-right` para posicionar a caixa de busca à esquerda ou à direita da barra de navegação respectivamente.

A marcação HTML típica para inserir uma caixa de busca em uma barra de navegação é mostrada em destaque a seguir.

- **HTML**

```
<nav class="navbar navbar-default">
...
   <div class="collapse navbar-collapse" id="barraComBusca">
      <form class="navbar-form navbar-right" role="search">
         <div class="form-group">
            <input type="text" class="form-control" placeholder="Search">
         </div>
         <button type="submit" class="
      </form>
   ...
</nav>
```

[.../c4/barra-navegacao-com-complementos.html]

4.7.4 Barra de navegação com botão

O Bootstrap prevê a classe `navbar-btn` a ser declarada no elemento `button` para inserir um botão na barra de navegação. Adicionalmente use as classes gerais `pull-left` e `pull-right` para posicionar o botão à esquerda ou à direita na barra de navegação respectivamente.

A marcação HTML típica para inserir um botão em uma barra de navegação é mostrada em destaque a seguir.

- **HTML**

```
<nav class="navbar navbar-default">
   ...
      </ul>
```

```
        <button type="button" class="btn btn-default navbar-btn pull-right">Login</button>
    </div><!-- /.navbar-collapse -->
</nav>
```

[.../c4/barra-navegacao-com-complementos.html]

4.7.5 Barra de navegação com link dropdown

A funcionalidade link dropdown foi estudada no item [4.6.3] e aplica-se as barras de navegação. Para estilizar dropdown está prevista uma marcação HTML padrão. A classe `dropdown` deve ser declarada para o elemento li que marca o link dropdown.

A marcação HTML típica para criar um link dropdown em uma barra de navegação é mostrada em destaque a seguir.

- **HTML**

```
<nav class="navbar">
...
    <li class="active"><a href="#">Home <span class="sr-only">(current)</span></a></li>
    <li><a href="#">Portfólio</a></li>
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
            aria-expanded="false">Artigos <span class="caret"></span></a>
            <ul class="dropdown-menu" role="menu">
                <li><a href="#">HTML</a></li>
                <li><a href="#">CSS</a></li>
                <li><a href="#">javascript</a></li>>Contato</a></li>
...
</nav>
```

[.../c4/barra-navegacao-com-complementos.html]

4.7.6 Barra de navegação com texto

Para inserir um texto em uma barra de navegação use o elemento `p` (parágrafo) e a ele defina a classe `navbar-text`. Use as classes gerais `pull-left` e `pull-right` para posicionar o botão à esquerda ou à direita na barra de navegação respectivamente.

A marcação HTML típica para criar textos em uma barra de navegação é mostrada em destaque a seguir.

- **HTML**

```
<nav class="navbar">
...
   <li class="active"><a href="#">Home <span class="sr-only">(current)</span></a></li>
   <li><a href="#">Portfólio</a></li>
   <li><a href="#">Portfólio</a></li>
</ul>
   <p class="navbar-text pull-right">Site criado pelo <a href="http://maujor.com">Maujor</a></p>
/div><!-- /.navbar-collapse -->
</nav>
```

[.../c4/barra-navegacao-com-complementos.html]

Observe na figura 4.8 os complementos para barras de navegação, conforme as marcações mostradas no itens [4.7.3], [4.7.4], [4.7.5] e [4.7.6].
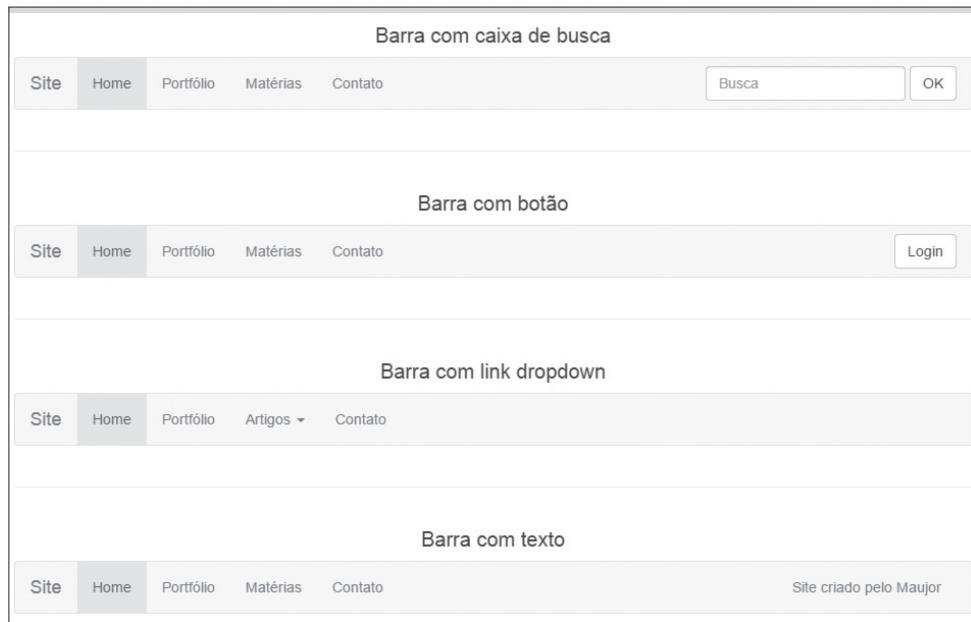
Figura 4.8 – Barra de navegação com complementos.

4.8 Caminho de navegação

Para criar um caminho de navegação, também conhecido pela sua designação em inglês *breadcrumb* (literalmente: migalhas de pão), usamos como container um elemento `ol` ao qual declaramos a classe `breadcrumb`. Os itens da lista marcam

os links no caminho e na sequência em que aparecem na marcação. Use a classe `active` para destacar o texto do link para a página atual.

A marcação HTML típica para criar um caminho de navegação é conforme mostrada no código a seguir.

- **HTML**

```
<ol class="breadcrumb">
   <li><a href="#">Serviços</a></li>
   <li><a href="#">active">Aplicações web</li>
</ol>
```

[.../c4/breadcrumb-paginacao-label-badge.html]

4.9 Paginação

Para criar um widget de paginação usamos como container geral o elemento `nav`. Dentro deste elemento usamos um elemento `ul` ao qual declaramos a classe `pagination`. Esse elemento é o container para uma lista contendo os números das páginas a navegar e os ícones para os links "Anterior" e "Próxima".

4.9.1 Widget de paginação básica

A marcação HTML típica para criar um widget de paginação básica é conforme mostrada no código a seguir.

- **HTML**

```
<nav>
   <ul class="pagination">
      <li><a href="#" aria-label="Anterior"><span aria-hidden="true">&laquo;</span></a></li>
      <li><a href="#">1</a></li>
      <li><a href="#">2</a></li>
      <li>a href="#">3</a></li>
      <li><a href="#">4</a></li>
      <li><a href="#">5</a></li>
      <li><a href="#" aria-label="Próxaquo;</span></a></li>
   </ul>
</nav>
```

[.../c4/breadcrumb -paginacao-label-badge.html]

Por padrão o widget de paginação é posicionado à esquerda. Declarar no container `nav` as classes `text-center` ou `pull-right` faz com que o widget seja posicionado no centro ou à direita respectivamente.

4.9.2 Dimensionamento do widget paginação

O widget de paginação tem dimensões padrão. Declarar no container `ul` as classes `pagination-lg` ou `pagination-sm` permite que se obtenha mais duas dimensões para o widget.

A marcação HTML típica para criar os três tamanhos de widget de paginação é conforme mostrada no código a seguir.

- **HTML**

```
<nav>
   <ul class="pagination pagination-lg"> <!-- tamanho maior -->
      <li><a href="#" aria-label="Anterior"><span aria-hidden="true">&laquo;</span></a></li>
...
```

ou

```
<nav>
   <ul class="pagination"> <!-- tamanho padrão -->
      <li><a href="#" aria-label="Anterior"><span aria-hidden="true">&laquo;</span></a></li>
...
```

ou

```
<nav>
   <ul class="pagination pagination-sm"> <!-- tamanho menor -->
      <li><a href="#" aria-label="Anterior"><span aria-hidden="true">&laquo;</span></
a></li>
...
```

4.9.3 Estado dos links

Os links que remetem às páginas são estilizados, por padrão, na cor branca no estado em repouso e na cor cinza clara no estado `mouseover`. O Bootstrap prevê as classes `active` e `disabled` a serem declaradas no elemento `li`, container do link, e que se destinam respectivamente a indicar visualmente a página corrente e a desabilitar um link.

A marcação HTML típica para aplicação dessas classes é conforme mostrada a seguir.

- **HTML**

```
<nav>
   <ul class="pagination">
      <li class="disabled"><a href="#" aria-label="Anterior"><span aria-hidden="true">
         &#171; span></a></li>
      <li><a href="#">1</a></li>
      <li><a href="#">2</a></li>
      <li class="active">a href="#">3</a></li>
      <li><a href="#">4</a></li>
      <li><a href="#">5</a></li>
      <li><a href="#" aria-laberue">&#187;</span></a></li>
   </ul>
</nav>
```

[.../c4/breadcrumb-paginacao-label-badge.html]

Notar o uso opcional das entidades HTML `«` para o link Anterior e `»` para o link Próximo em lugar de `«` e `»` respectivamente.

4.9.4 Pager

Os links que remetem às páginas anterior e próxima são chamados de pager. O Bootstrap prevê a classe `pager` a ser declarada no elemento `ul`, container dos dois links. Os links são estilizados como botões com bordas arredondadas e por padrão são posicionados no centro da página. O container geral do pager deve ser um elemento `nav`.

É possível posicionar um link à esquerda e outro à direita do container com uso das classes `previous` e `next` respectivamente no elemento `li` container do link. Para desabilitar um link use a classe `disabled` no elemento `li`.

A marcação HTML típica para criar o widget pager é conforme mostrada no código a seguir

- **HTML**

```
<nav>
   <ul class="pager">
      <li><li><a href="#">Próxima</a></li>
   </ul>
</nav>
```

[.../c4/breadcrumb-paginacao-label-badge.html]

4.10 Rótulos e marcadores

Para criar rótulos estilizados com seis cores diferentes o Bootstrap prevê a declaração da classe de estilização geral `label` e das classes para colorir `label-default`, `label-primary`, `label-success`, `label-info`, `label-warning` e `label-danger` a serem declaradas em um elemento `span` contendo o rótulo.

Para criar marcadores o Bootstrap prevê a declaração da classe de estilização `badge` a ser declarada em um elemento `span` contendo o marcador. Por padrão o rótulo é inserido em um pequeno retângulo cinza com bordas arredondadas e na cor branca.

A marcação HTML típica para criar rótulos e marcadores é conforme mostrada no código a seguir:

- **HTML**

```
<nav> <!-- Rótulos -->
   <ul class="pager">
      <li><a href="#">Anterior</a></li>
      <li><a href="#">Próxima</a></li>
   </ul>
</nav>
<h1>Artigosh1>
<h6>Menssagens <span class="badge">358</span></h6>
```

[.../c4/breadcrumb-paginacao-label-badge.html]

Observe na figura 4.9 os widgets, conforme as marcações mostradas no itens [4.8], [4.9], [4.10] e [4.11].
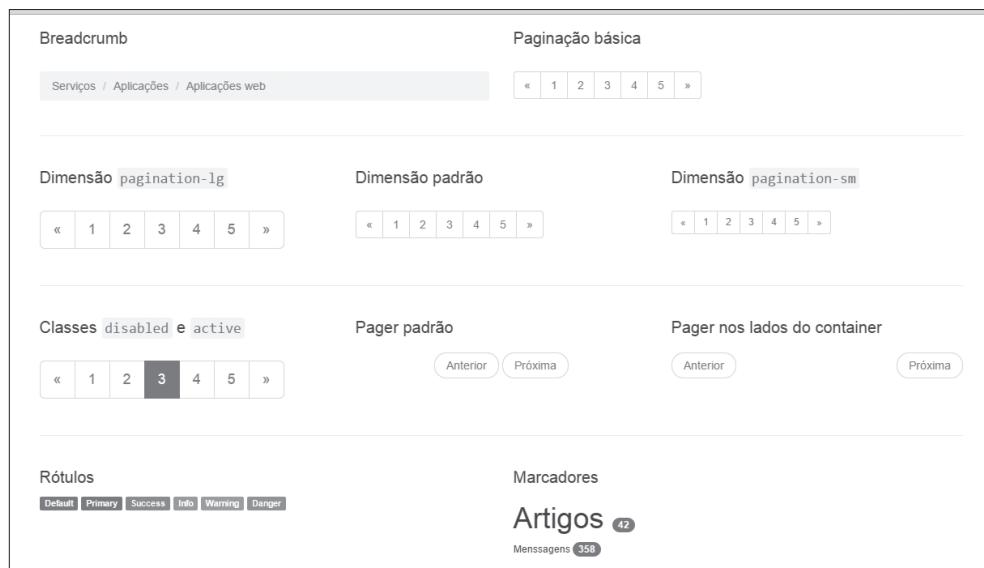
Figura 4.9 – Widgets.

4.11 Jumbotron

A palavra Jumbo Tron foi inventada pela Sony para designar uma invenção que conhecemos como "Telão", presentes em locais de grandes eventos como campos de futebol e estádios para shows.

No Bootstrap o widget jumbotron é uma espécie de "telão" estilizado por padrão com bordas e fundo cinzas destinado a destacar conteúdos.

É possível obter-se dois tipos de jumbotron conforme mostrados a seguir.

4.11.1 Jumbotron básico

Jumbotron básico é aquele cuja largura é igual à largura do seu container na página, ou seja, não se estende necessariamente por toda a largura da tela. Para criar um jumbotron usa-se um elemento container com a classe `jumbotron`.

A marcação HTML típica para criar o widget jumbotron básico é conforme mostrada no código a seguir:

- **HTML**

```
...
<div class="jumbotron">
   <h1>Jumbotron básico</h1>
   <p>Lorem ipsum...</ btn-lg" href="#" role="button">Saiba mais</a></p>
</div>
```

[.../c4/jumbotron.html]

4.11.2 Jumbotron estendido

Jumbotron estendido é aquele cuja largura é igual à largura da tela. Para obter este tipo de jumbotron ele deve ser marcado fora do `div.container` que contém toda a marcação do Boostrap, ou seja, deverá ser elemento filho de `body`. Além disso, deverá ser criado um `div.container` elemento-filho de `div.jumbotron`.

A marcação HTML típica para criar o widget jumbotron estendido é conforme mostrada no código a seguir:

- **HTML**

```
...
<div class="jumbotron">
<div class="container">
   <h1>Jumbotron estendido</h1>
   <p>Lorem ipsum...</p>
   <pg" href="#" role="button">Saiba mais</a></p>
</div>
</div>
```

[.../c4/jumbotron.html]

Observe na figura 4.10 os widgets, conforme as marcações mostradas no itens [4.11.1e [4.11.2].
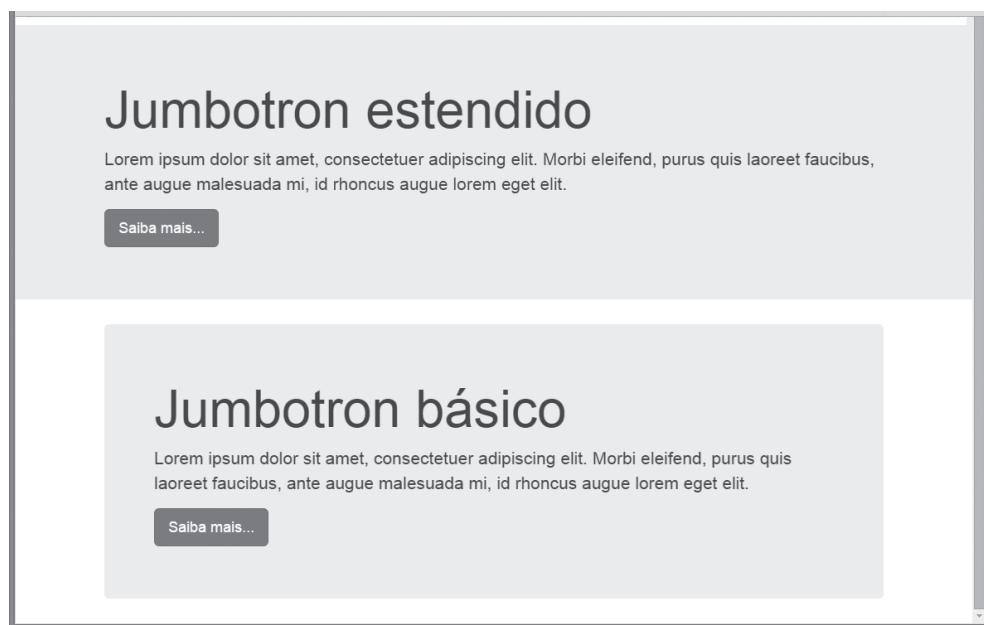
Figura 4.10 – Jumbotron.

4.12 Page-header

O Bootstrap prevê a classe `page-header` para ser definida em um elemento container para um título. O efeito desta classe é o de estilizar o título com uma borda inferior e margens.

A marcação HTML típica para criar estilizar títulos com uso de page-header é conforme mostrada no código a seguir:

- **HTML**

```
<div class="page-header">
  ge-header</code> <small> com texto menor</small></h1>
</div>
```

[.../c4/jumbotron.html]

O exemplo desta funcionalidade encontra-se na mesma página que demonstra o jumbotron.

4.13 Thumbnail

Thumbnail é uma palavra inglesa que em tradução livre significa unha do polegar, mas é usada também para designar uma coisa curta ou pequena. E é com este sentido que ela se aplica ao jargão empregado em desenvolvimento web. Em resumo no nosso contexto seu significado é: *miniatura*.

Normalmente um thumbnail é uma miniatura de uma imagem, vídeo, texto ou um conteúdo qualquer que quando clicada leva o usuário à sua forma em tamanho real. O exemplo padrão e mais comum do uso de thumbnails na web é aquele no qual é apresentado ao usuário um grid de imagens em miniaturas que quando clicadas acessam a imagem em seu tamanho real.

4.13.1 Thumbnail padrão

O Bootstrap prevê a classe `thumbnail` para ser definida em um elemento âncora a que servirá de container para o elemento `img` que contém o thumbnail. Para dispor os thumbnails lada a lado use o grid.

A marcação HTML típica para criar um grid de thumbnails é conforme mostrada no código a seguir:

- **HTML**

```
<div class="col-md-3 col-sm-6">
   <a href="t1.jpg" class="thumbnail"><img src="t1.jpg" alt="carro1"></a>
</div>
<div class="col-md-3 col-sm-6">
   <a href="t2.jpg " class="thumbnail"><img src="t2.jpg" alt=" carro2"></a>
</div>
<div class="col-md-3 col-sm-6">
   <a href="t3.jpg " class="thumbnail"><img src="t3.jpg" alt=" carro3"></a>
</div>
<div class="col-md class="thumbnail"><img src="t4.jpg" alt=" carro4"></a>
</div>
```

[.../c4/thumbnail.html]

4.13.2 Thumbnail com conteúdos

O Bootstrap oferece funcionalidades para criação de thumbnails com conteúdos que resultam em widgets com finalidades diferentes daquela de simplesmente lincar para uma imagem em tamanho real.

O exemplo mais comum de um desses widgets é aquele usado em uma página de um site de vendas online contendo thumbnails mostrado a imagem de um produto e a seguir textos designando e descrevendo o produto, com link para informações adicionais.

Existe a opção de inserir conteúdo no thumbnail. Neste caso retira-se o link da imagem do thumbnail e cria-se um container para os conteúdos a inserir. Esse container deverá receber a classe `caption`.

A marcação HTML típica para criar um thumbnail com conteúdos é idêntica à marcação mostrada anteriormente com as alterações conforme mostrada no código a seguir:

- **HTML**

```
<div class="col-md-3 col-sm-6">
<div class="thumbnail">
   <a href="t1.jpg" class="thumbnail"><img src="t1.jpg" alt="carro1"></a>
      <div class="caption">
         <h3>Descrição da imagem</h3>
         <p>Lorem ipsum dolor...</p>
   href="#" role="button">Saiba mais...</a></p>
      </div>
</div>
...
```

[.../c4/thumbnail.html]

Observe na figura 4.11 os thumbnails conforme as marcações mostradas.
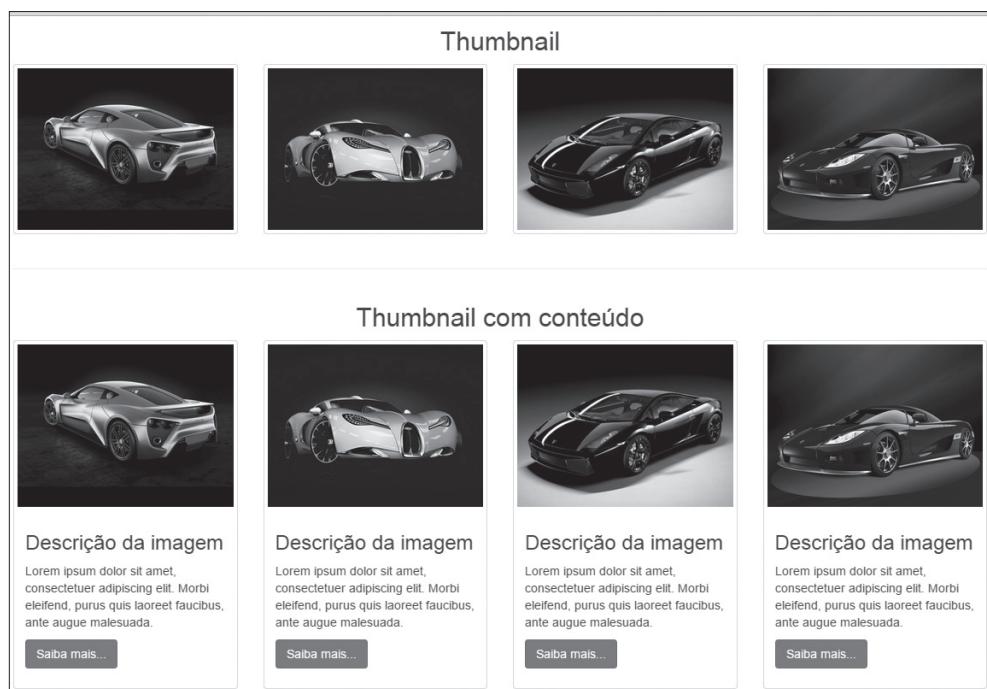
Figura 4.11 – Thumbnail.

4.14 Caixas de alerta

O Boostrap prevê a criação de caixas contendo mensagem ao usuário. Essas caixas podem ser estilizadas de quatro formas conforme a natureza da mensagem (mensagem de erro, mensagem de sucesso, mensagem de alerta e mensagem informativa).

As caixas de mensagem têm bordas arredondadas, fundo e texto estilizados em tons de cor conforme a natureza da mensagem.

A marcação típica para criar uma caixa de alerta consta de um `div` container ao qual se atribui a classe `alert` e o par atributo/valor `role="alert"`.

As classes `alert-success`, `alert-info`, `alert-warning` e `alert-danger` a serem marcadas no container estilizam a caixa do alerta com a cor apropriada de sucesso (verde), informação (azul), alerta (amarela) e erro (salmão).

Para criar um botão para fechar a caixa de alerta usa-se a classe `alert-dismissible` no cantainer e dentro dele marca-se um elemento `button` do com a classe `close`, o par atributo/valor `data-dismiss="alert"`.

Para criar um link dentro da caixa de alerta use a classe `alert-link` no elemento a que marca o link.

A marcação HTML típica para criar uma caixa de alerta com todas as funcionalidades descritas é conforme mostrada no código a seguir:

- **HTML**

```
<div class="alert alert-success alert-dismissible" role="alert">
   <button type="button" class="close" data-dismiss="alert" aria-label="Fechar">
      <span aria-hidden="true">&times;</span>
   </button>
   <p><strong>Atenuma
      <a href="#" class="alert-link">mensagem de alerta.</a>
   </p>
</div>
```

[.../c4/caixa-de-alerta.html]

4.15 Barra de progresso

Barra de progresso é um widget destinado a oferecer uma indicação visual ao usuário do andamento de uma tarefa tal como o download de um arquivo.

Este widget usa as funcionalidades de transição e animação das CSS3 que não são suportadas pelos IE9 e anteriores bem como por versões antigas do demais navegadores navegadores.

A marcação típica para criar uma barra de progresso básica consta de um `div` container ao qual se define a classe `progress` e um `div` elemento-filho ao qual se define os seguintes atributos com seus respectivos valores:

Atributo	Valor – Finalidade
`class`	progress-bar – Servir de âncora para estilização.
`role`	progressbar – Atributo destinado a definir o valor semântico do elemento `div`.
`aria-valuenow`	número – Indica o valor atual do progresso da tarefa.
`aria-valuemin`	número – Indica o valor mínimo medido pela barra.
`aria-valuemax`	número – Indica o valor máximo medido pela barra.
`style`	width em porcentagem – Indica visualmente o valor do progresso da tarefa.

Os atributos `role` e `aria-*` visam a incrementar a acessibilidade da barra de progresso.

A porcentagem de progresso na barra é obtida com declaração inline para a propriedade `width`, expressa em porcentagem.

Declara-se o progresso da tarefa dentro de um elemento `span` ao qual se atribui a classe `sr-only` com a finalidade de fazê-lo acessível somente à tecnologias assistivas.

A marcação para criar uma barra de progresso básica é conforme mostrado a seguir.

- **HTML**

```
<div class="progress">
   <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0"
      aria-valuemax="100" style="width: 60%;">
   <span class="sr-only">60% Complete</span>
  </div>
</div>
```

Essa marcação renderiza uma barra de progresso padrão cuja faixa indicativa do progresso da tarefa é estilizada em gradiente na cor azul escura.

Para obter barras de progresso estilizadas nas cores azul claro, verde, laranja e vermelho basta declarar no `div.progress-bar` as classes `progress-bar-info`, `progress-bar-success`, `progress-bar-warning` e `progress-bar-danger`, respectivamente.

Para obter barras de progresso estilizadas segundo o efeito zebra, basta declarar no `div.progress-bar` a classe `progress-bar-striped`.

Para obter barras de progresso estilizadas segundo o efeito zebra e animadas, basta declarar no `div.progress-bar` as classes `progress-bar-striped` e `active`.

Para obter barras de progresso múltiplas, basta acrescentar elementos-filho no container da barra.

A marcação HTML mostrada a seguir esclarece a obtenção desses diferentes tipos de barra de progresso:

- **HTML**

```
<div class="row">
<div class="col-md-6">
   <h4>Barras de progresso com cor em gradiente</h4>
   <div class="progress">
      <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0"
          aria-valuemax="100" style="width: 60%;">
```

```
            <span class="sr-only">60% Complete</span>
         </div>
      </div>
      <div class="progress">
         <div class="progress-bar progress-bar-info" role="progressbar" aria-valuenow="40"
               aria-valuemin="0" aria-valuemax="100" style="width: 40%;">
            <span class="sr-only">40% Complete</span>
         </div>
      </div>
      <div class="progress">
         <div class="progress-bar progress-bar-success" role="progressbar"
            aria-valuenow="80" aria-valuemin="0" aria-valuemax="100" style="width: 80%;">
            <span class="sr-only">80% Complete</span>
         </div>
      </div>
      <div class="progress">
         <div class="progress-bar progress-bar-warning" role="progressbar"
            aria-valuenow="30" aria-valuemin="0" aria-valuemax="100" style="width: 30%;">
            <span class="sr-only">30% Complete</span>
         </div>
      </div>
      <div class="progress">
         <div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="50"
            aria-valuemin="0" aria-valuemax="100" style="width: 50%;">
            <span class="sr-only">50% Complete</span>
         </div>
      </div>
   <div class="col-md-6">
      <h4>Barra de progresso com efeito zebra</h4>
      <div class="progress">
         <div class="progress-bar" role="progressbar progress-bar-striped"
            aria-valuenow="60" aria-valuemin="0" aria-valuemax="100" style="width: 60%;">
            <span class="sr-only">60% Complete</span>
         </div>
      </div>
      <div class="progress">
         <div class="progress-bar progress-bar-info progress-bar-striped" role="progressbar"
            aria-valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width: 40%;">
            <span class="sr-only">40% Complete</span>
         </div>
```

```
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-success progress-bar-striped"
        role="progressbar" aria-valuenow="80" aria-valuemin="0" aria-valuemax="100"
        style="width: 80%;">
        <span class="sr-only">80% Complete</span>
      </div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-warning progress-bar-striped"
        role="progressbar" aria-valuenow="30" aria-valuemin="0" aria-valuemax="100"
        style="width: 30%;">
        <span class="sr-only">30% Complete</span>
      </div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-danger progress-bar-striped"
        role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"
        style="width: 50%;">
        <span class="sr-only">50% Complete</span>
      </div>
    </div>
  </div>
  </div> <!-- /.row -->
  <div class="row">
  <div class="col-md-6">
    <h4>Barra de progresso animada</h4>
    <div class="progress">
      <div class="progress-bar progress-bar-danger progress-bar-striped active"
        role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"
        style="width: 50%;">
        <span class="sr-only">50% Complete</span>
      </div>
    </div>
  <div class="col-md-6">
  <h4>Barra de progresso múltipla</h4>
    <div class="progress">
      <div class="progress-bar progress-bar-danger progress-bar-striped "
        role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-valuemax="100"
        style="width: 40%;">
```

```
            <span class="sr-only">40% Complete</span>
        </div>
        <div class="progress-bar progress-bar-info" role="progressbar" aria-valuenow="35"
            aria-valuemin="0" aria-valuemax="100" style="width: 35%;">
            <span class="sr-only">35% Complete</span>
        </div>
        <div class="progress-bar progress-bar-warning" role="progressbar"
            aria-valuenow="10" aria-valuemin="0" aria-valuemax="100" style="width: 10%;">
            <span class="sr-only">10% Complete</span>
        </div>
</div>
</div>
</div> <!-- /.row -->
<div class="row">
<div class="col-md-6">
    <h4>Barras de progresso com rótulo</h4>
    <div class="progress">
        <div class="progress-bar" role="progressbar" aria-valuenow="6ax="100"
            style="width: 60%;">
            60%
        </div>
    </div>
</div>
</div> <!-- /.row -->
```

[.../c4/barra-de-progresso.html]

Na figura 4.12, mostramos a renderização das barras de progresso, conforme marcação HTML anterior.
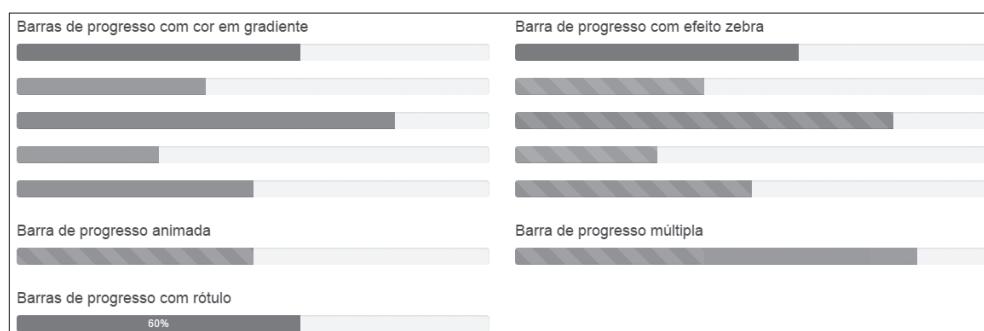
Figura 4.12 – Barras de progresso.

4.16 Objetos de mídia

As funcionalidades do Bootstrap para marcação de objetos de mídia destinam-se a criar componentes do tipo cometários em blogs, tweets etc. cujo padrão de apresentação baseia-se na inserção de uma imagem à esquerda ou à direita de um conteúdo textual descritivo da mídia.

A marcação padrão para esse componente usa três elementos `div` conforme descritos a seguir.

- Elemento `div` container geral ao qual atribui-se a classe `media`.
- Elemento-filho `div` ao qual atribui-se a classe `media-body` destinado a marcar conteúdos textuais que se destinam a descrever a mídia. O elemento `h1-h6` que marca o título do texto descritivo deverá receber a classe `media-heading`.
- Elemento-filho `div` destinado a servir de container para a mídia e ao qual atribui-se a classe `media-left` ou `media-right` conforme se queira posicionar a mídia à esquerda ou à direita do conteúdo descritivo da mídia. Este elemento deverá ser posicionado na marcação antes do `div.media-body` se o alinhamento for à esquerda ou depois do `div.media-body` se o alinhamento for à direita. Ao elemento que marca a mídia (`img`, `video`, `canvas` etc.) define-se a classe `media-object`.

Por padrão o topo da mídia é alinhado com o topo do texto descritivo. Podemos alterar este alinhamento definindo as classes `media-middle` ou `media-bottom` no `div.media-body`. Estas classes se destinam a alinhar verticalmente a mídia no centro ou na parte inferior, respectivamente, do texto descritivo.

O exemplo mostrado a seguir esclarece a marcação HTML e o uso destas classes.

- **HTML**

```
<div class="media"> <!-- mídia à esquerda -->
   <div class="media-left" href="#">
      <img class="media-object" src="64x64.gif" alt="">
   </div>
   <div class="media-body">
      <h4 class="media-heading">Título</h4>
      <p>...</p>
   </div>
</div>
<div class="media"> <!-- mídia à direita -->
   <div class="media-body">
      <h4 class="media-heading">Título</h4>
```

```
        <p>...</p>
    </div>
    <div class="media-right" href="#">
        <img class="media-object" src="64x64.gif" alt="">
    </div>
</div>
<div class="media"> <!-- alinhamento vertical no meio -->
    <div class="media-left media-middle" href="#">
        <img class="media-object" src="64x64.gif" alt="">
    </div>
    <div class="media-body">
        <h4 class="media-heading">Título</h4>
        <p>...</p>
    </div>
</div>
<div class="media"> <!-- alinhamento vertical inferior -->
    <div class="media-left media-bottom" href="#">
        <img class="media-object" srdiv class="media-body">
        <h4 class="media-heading">Título</h4>
        <p>...</p>
    </div>
</div>
```

[.../c4/objetos-de-midia-1.html]

Na figura 4.13, mostramos a renderização dos objetos de mídia, conforme a marcação HTML mostrada anteriormente.

Figura 4.13 – Alinhamento de objetos de mídia.

Opcionalmente podemos inserir objetos de mídia com uso de elementos para marcação de listas que podem ser aninhados, conforme a estrutura HTML geral mostrada a seguir.

- **HTML**

```
<ul class="media-list">
   <li class="media">
      <div class="media-left" href="#">
         <img class="media-object" src="64x64.gif" alt="">
      </div>
      <div class="media-body">
      <h4 class="media-heading">Título</h4>
      <p>...</p>
      <div class="media"> <!-- mídia aninhada -->
         <div class="media-left" href="#">
            <img class="media-object" src="64x64.gif" alt="a-body">
         <h4 class="media-heading">Título aninhado</h4>
         <p>...</p>
      ...
   </li>
</ul>
```

[.../c4/objetos-de-midia-2.html]

Na figura 4.14, mostramos a renderização dos objetos de mídia aninhados, conforme a marcação HTML mostrada anteriormente.
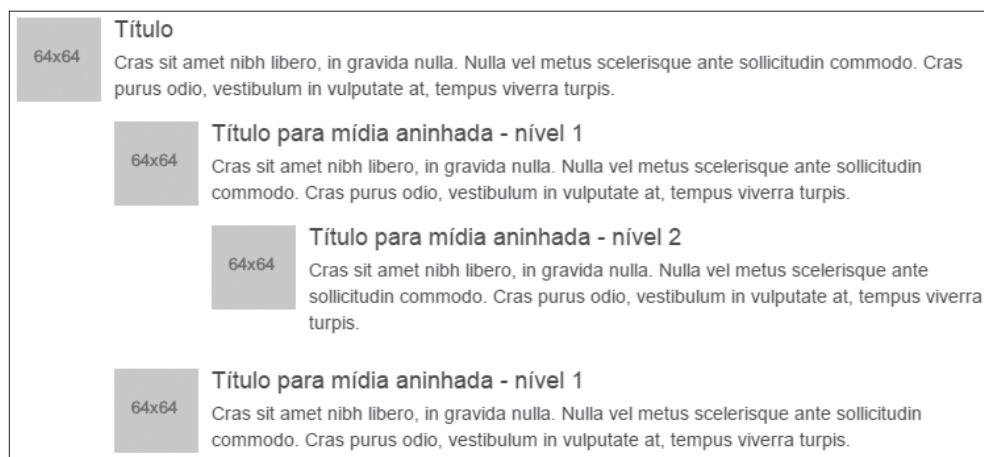
Figura 4.14 – Objetos de mídia aninhados.

4.17 Agrupamento de itens de listas

Denomina-se agrupamento de itens de listas um componente do Bootstrap cuja finalidade é criar estilização diferenciada para as listas da HTML.

A marcação HTML para criar uma apresentação básica deste componente é conforme mostrada a seguir.

- **HTML**

```
<ul class="list-group">
   <li class="list-group-item">Abacate</li>
   <li class="list-group-item">Carambola</li>
   <li class="list-group-item">Figo</li>
   <li class="list-group-item">Uva</li>
</ul>
```

É possível inserir um “badge” (termo inglês que neste contexto significa marcador e se destina a fornecer informação adicional) em cada item da lista. Por padrão o identificador é posicionado à direita do texto do item da lista.

A marcação HTML para criar um marcador em itens da lista é conforme mostrada a seguir.

- **HTML**

```
<ul class="list-group">
   <li class="list-group-item"> <span class="badge">14</span> Abacate</li>
   <li class="list-group-item"> <span class="badge">fora de época</span> Carambola</li>
   <li class="list-group-item"> <span class="badge">345</span> Figo</li>
   <li class="list-group-item"> <span class="badge">em falta</span> Uva</li>
</ul>
```

Para criar uma lista de links use um container `div` em lugar do elemento `ul` e elementos `a` em lugar de `li` conforme marcação HTML mostrada a seguir.

- **HTML**

```
<div class="list-group">
   <a class="list-group-item active" href="#">Site do Maujor</a>
   <a class="list-group-item" href="#">Wikipedia português</a>
   <a class="list-group-item disabled" href="#">Site do W3C</a>
   <a class="list-group-item" href="#">Google</a>
</div>
```

Para criar uma lista de botões com efeito rollover use um container `div` em lugar do elemento `ul` e elementos `button` em lugar de `li` conforme marcação HTML mostrada a seguir.

- **HTML**

```
<div class="list-group">
   <button type="button" class="list-group-item">Arquivo</button>
   <button type="button" class="list-group-item">Inserir</button>
   <button type="button" class="list-group-item">Copiar</button>
   <button type="button" class="list-group-item">Enviar</button>
</div>
```

As classes contextuais se destinam a estilizar um item da lista nas cores verde, azul claro, amarela e vermelha, cor esta que transmite a ideia do contexto do item. As classes para definir as cores citadas são `list-item-success`, `list-item-info`, `list-item-warning` e `list-item-danger`, respectivamente, conforme a marcação HTML mostrada a seguir.

- **HTML**

```
<ul class="list-group">
   <li class="list-group-item list-group-item-success">Arquivo enviado!</li>
   <li class="list-group-item list-group-item-info">Use o formato JPG</li>
   <li class="list-group-item list-group-item-warning">Está faltando um arquivo</li>
   <li class="list-group-item list-group-item-danger">Poderá conter virus</li>
</ul>
```

Para desabilitar um item da lista use a classe `disabled` no elemento que marca o item. Itens desabilitados são renderizados com fundo na cor cinza. No exemplo que mostra a lista de links o item "Site do W3C" foi desabilitado.

Dica: As classes contextuais e as classes `active` e `disabled` podem ser usadas em qualquer um dos agrupamentos de itens mostrados anterioos diferentes agrupamentos de itens de lista conforme as marcações HTML anteriores encontra-se em [*.../c4/agrupamento-de-itens-de-lista.html*].

Na figura 4.15, mostramos a renderização destes agrupamentos de itens de lista.

Figura 4.15 – Agrupamento de itens de lista.

4.18 Painéis

Painel é o componente do Bootstrap destinado a criar um box para conteúdos. O box é estilizado com bordas em cantos arredondados e seu fundo e conteúdos recebem estilização definida por classes específicas.

A marcação HTML para criar um painel básico é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-body">
      <p>Conteúdo textual do painel básico - Lorem ...</p>
   </div>
</div>
```

A marcação HTML para criar um painel com título é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">Título do texto do painel</h3>
   </div>
   <div class="panel-body">
```

```
      <p>Conteúdo textual do painel básico - Lorem ...</p>
   </div>
</div>
```

A marcação HTML para criar um painel com rodapé é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-body">
      <p>Conteúdo textual do painel básico - Lorem ...</p>
   </div>
   <div class="panel-footer">Rodape do painel</div>
</div>
```

Para criar um painel com título e bordas estilizadas nas cores contextuais, azul escuro, verde, azul claro, amarela e vermelha, cor esta que transmite a ideia do contexto usamos, no container geral do painel, as classes `panel-primary`, `panel-success`, `panel-info`, `panel-warning` e `panel-danger` respectivamente conforme mostrado na marcação a seguir.

- **HTML**

```
<div class="panel panel-primary">
   <div class="panel-heading">
      <h3 class="panel-title">Título do texto do painel</h3>
   </div>
   <div class="panel-body">
      <p>Conteúdo textual do painel básico</p>
   </div>
</div>
```

A marcação HTML para criar um painel *com* conteúdo textual contendo uma tabela é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">Título do painel com tabela</h3>
   </div>
   <div class="panel-body">
      <p>Conteúdo textual do painel com uma tabela - Lorem...</p>
```

```
   </div>
   <table class="table">
      ...
   </table>
</div>
```

A marcação HTML para criar um painel *sem* conteúdo textual contendo uma tabela é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">Título do painel com tabela</h3>
   </div>
<table class="table">
      ...
   </table>
</div>
```

A marcação HTML para criar um painel contendo um agrupamento de itens é conforme mostrada a seguir.

- **HTML**

```
<div class="panel panel-default">
   <div class="panel-heading">
      <h3 class="panel-title">Título do painel com agrupamento de itens</h3>
   </div>
   <div class="panel-body">
      <p>Conteúdo textual do painel com um agrupamento de itens</p>
   </div>
   <ul class="list-group">
      <li class="list-group-item">Abacate</li>
      <li class="list-group-item">Carambola</li>
      <li class="list-group-item">Figo</li>
      <li class="list-group-item">Uva</li>
   </ul>
</div>
```

Dica: Tal como mostrado para painel com tabelas, o painel com agrupamento de itens poderá conterrquivo que demonstra os diferentes tipos de painel conforme as marcações HTML anteriores encontra-se em [*.../c4/painel.html*].

Na figura 4.16, mostramos a renderização destes tipos de painel.
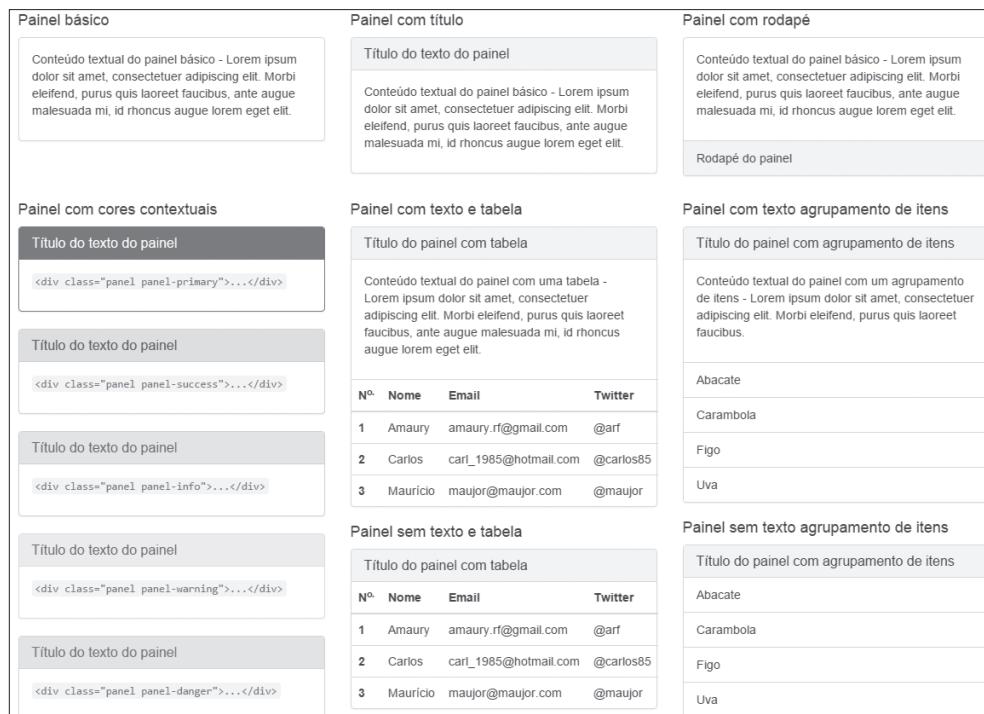
Figura 4.16 – Painéis.

4.19 Mídias responsivas

O Bootstrap prevê classes para tornar responsiva a incorporação de conteúdos de terceiros, ou próprios, com uso dos elementos de marcação `iframe`, `embed`, `video` e `object`.

A marcação HTML para tornar responsiva a incorporação com `iframe` é conforme mostrada a seguir.

- **HTML**

```
<!-- Mídias com razão de aspecto 16:9 -->
<div class="embed-responsive embed-responsive-16by9">
   <iframe class="embed-responsive-item" src="..."></iframe>
</div>
<!-- Mídias com razão de aspecto 4:3 -->
<div class="embed-responsive embed-responsive-4by3">
   <iframe class="embed-responsive-item" src="..."></iframe>
</div>
```

Para incorporar um vídeo próprio com uso do elemento `video` da HTML5 a marcação HTML é conforme mostrada a seguir.

- **HTML**

```
<div class="embed-responsive embed-responsive-4by3">
   <figure class="embed-responsive-item">
      <video poster="cover.jpg" controls>
         <source src="plymouth.ogg" type="video/ogg" />
         <source src="plymouth.mp4" type="video/mp4" />
         <source src="plymouth.webm" type="video/webm" />
         <p>Lamento! Seu navegador não suporta o elemento vo video localmente <br />
         <a href="plymouth.mp4">fazendo o download</a></p>
      </video>
   </figure>
</div>
```

[.../c4/video-html5-responsivo.html]

Para incorporar um vídeo do YouTube a marcação HTML é conforme mostrada a seguir.

- **HTML**

```
<div class="embed-responsive embed-responsive-16by9">
   <figure class="embed-responsive-item">
      <iframe src="https://www.youtube.com/embed/90R6NNXEIsE" allowfullscreen></iframe>
   </figure>
</div>
```

[.../c4/video-youtube-responsivo.html]

Dica: O código de incorporação (elemento `iframe`) é fornecido pelo YouTube e o Bootstrap permite que se retire do código os atributos `width`, `height` e `frameborder`.

Para incorporar um mapa do Google a marcação HTML é conforme mostrada a seguir.

- **HTML**

```
<div class="embed-responsive embed-responsive-4by3">
   <figure class="embed-responsive-item">
      <iframe src=" https://www.google.com/maps/embed?..." allowfullscreen></iframe>
   </figure>
</div>
```

[.../c4/mapa-google-responsivo.html]

Dica: O código de incorporação (elemento `iframe`) é fornecido pelo Google Maps e o Bootstrap permite que se retire do código os atributos `width`, `height, style` e `frameborder`.

4.20 Wells

Este é um componente do Bootstrap cuja finalidade é criar um box, com `padding` padrão, para um elemento da marcação estilizado com uma borda de cantos arredondados, uma cor de fundo e um efeito de baixo-relevo.

Basta que se defina a classe `well` no elemento, para que se dê a estilização. As classes opcionais `well-sm` e `well-lg` permitam que se diminua ou aumente respectivamente o `padding` padrão obtendo boxes com altura menor e maior em relação a altura padrão.

A marcação HTML para criar os três tipos de box (três alturas) é conforme mostrada a seguir.

- **HTML**

```
<p class="well well-sm">Pellentesque sapien...</p>
<p class="well">Pellentesque sapien...</p>
<p class="well well-lg">Pellentesque sapien...</p>
```

[.../c4/wells.html]

Na figura 4.17, mostramos a renderização dos três tipos de box.

Well altura reduzida

Pellentesque sapien. Sed purus sapien, eleifend ac, volutpat ac, molestie eget, massa. Nam a nibh. Nulla facilisi.

Well padrão

Pellentesque sapien. Sed purus sapien, eleifend ac, volutpat ac, molestie eget, massa. Nam a nibh. Nulla facilisi.

Well altura aumentada

Pellentesque sapien. Sed purus sapien, eleifend ac, volutpat ac, molestie eget, massa. Nam a nibh. Nulla facilisi.

Figura 4.17 – Wells.

CAPÍTULO 5

Plugins

Vimos no capítulo anterior que alguns componentes do Bootstrap são interativos. Consideremos a criação de menu dropdown conforme estudado no item [4.2]. Naquele item mencionamos, sem mais considerações, que a condição para o funcionamento do menu (abertura e fechamento) era a inclusão, na página do menu, da biblioteca jQuery, do plugin dropdown e da definição do par atributo/valor `data-*`. Naquele capítulo mencionamos ainda a necessidade de inclusão de plugins para fazer funcionar a navegação em abas e as caixas de mensagens. Plugins estão incluídos no pacote de download do Bootstrap e eles não são apenas aqueles citados no capítulo 4. Neste capítulo estudaremos os plugins do Bootstrap suas aplicações e uso.

5.1 Introdução

Quando fazemos o download do Bootstrap (ver item [1.3]) obtemos uma pasta denominada *js* que contém dois arquivos: o arquivo com *todos* os plugins do Bootstrap em versão comentada e versão comprimida.

Todos os componentes interativos do Boostrap dependem da inclusão da biblioteca jQuery e do arquivo de plugins para funcionar.

É quase certeza que um projeto não utilize *todos* os plugins do Bootstrap. Nesse caso seria injustificável lincar para o arquivo obtido no download do Bootstrap e seria mais sensato fazer o download personalizado, incluindo somente os plugins a usar no projeto.

A marcação HTML típica para incluir os arquivos JavaScript que farão funcionar a interação na página é conforme mostrado a seguir.

- **HTML**

```
<!-- Scripts para fazer funcionar o menu -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../bootstrap/js/jquery.min.js">
<\/script>')</script>
<script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
<body>
<html>
```

Nessa marcação optamos por incluir os arquivos imediatamente antes da tag de fechamento do elemento `body` e não é do escopo deste livro discutir o local de inclusão de scripts na página.

Convém esclarecer dois pontos importantes sobre o código mostrado:

- Na data em que escrevemos este capítulo a versão mais recente da biblioteca jQuery era a 1.11.2. No seu projeto altere para a versão atual.
- Salve uma cópia da biblioteca jQuery na pasta *js* (que você obteve com o download) para criar o link que servirá de fallback para o caso de falha no CDN Google. No nosso caso salvamos a biblioteca com o nome de arquivo *jquery.min.js*.

O Bootstrap nos oferece duas opções para aplicar o método principal de um plugin em um elemento da marcação. Uma opção é com a definição, diretamente na marcação HTML, de um atributo `data-*` no elemento que recebe o método do plugin e a outra é com uso da sintaxe da biblioteca jQuery para aplicação de seus métodos. Vejamos cada uma dessas opções tomando como exemplo um link de um menu dropdown.

Em [4.1] estudamos a construção de menus dropdown e vimos que eles dependem do plugin dropdown. Para mostrar as duas opções de fazer funcionar o plugin, considere a marcação HTML retirada daquele item, para o link Frutas.

- **HTML** – Plugin com uso do atributo data-*

```
...
<div class="dropdown">
<button class="btn btn-primary dropdown-toggle" type="button" id="dropDown1"
   data-toggle="dropdown" aria-expanded="true">Frutas <span class="caret"></span></button>
   <ul class="dropdown-menu" role="menu" aria-labelledby="dropDown1">
      <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Abacaxi</a></li>
```

```
        <li role="presentation">...</li>
        ...
    </ul>
</div> <!-- /.dropdown -->
<!-- conteúdos da página -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../bootstrap/js/jquery.min.js"><\/
script>')</script>
<script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
<body>
<html>
```

Nessa opção definimos o atributo `data-toggle="dropdown"` no elemento que aciona o submenu e o Bootstrap com seus mecanismos internos aplica o método `dropdown()` do plugin naquele elemento.

- **HTML** – Plugin com uso da sintaxe jQuery

```
...
<div class="dropdown">
<button class="btn btn-primary dropdown-toggle" type="button" id="dropDown1"
    data-toggle="dropdown" aria-expanded="true">Frutas <span class="caret"></span></button>
    <ul class="dropdown-menu" role="menu" aria-labelledby="dropDown1">
        <li role="presentation"><a role="menu-item" href="#" tabindex="-1">Abacaxi</a></li>
        <li role="presentation">...</li>
        ...
    </ul></div> <!-- /.dropdown -->
<!-- conteúdos da página -->
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../bootstrap/js/jquery.min.js">
<\/script>')</script>
<script type="text/javascript" src="../bootstrap/js/bootstrap.min.js"></script>
    <script type="text/javascript">
        $('.dropdown-toggle').dropdown();
    </script>
<body>
<html>
```

Nessa opção não há definição de atributo `data-toggle="dropdown"` no elemento que aciona o menu. Usa-se a sintaxe jQuery padrão para chamar o método `dropdown()` naquele elemento.

O uso do atributo `data-*` deve ser considerado em primeiro lugar, contudo a outra opção oferece ao desenvolvedor maior flexibilidade na aplicação do método, podendo tirar proveito, por exemplo, do encadeamento que a sintaxe jQuery possibilita como mostrado a seguir.

```
$('.dropdown-toggle').dropdown().css('color', '#f00');
```

ou ainda, definir opções:

```
$('#janelaModal').modal( { keyboard: false } ); // desabilita abertura por teclado
```

É possível desabilitar globalmente a opção `data-*` para fazer funcionar plugins conforme mostrado a seguir:

```
$('body').off('data-api'); // desabilita para todos os plugins
```

Podemos também desabilitar a opção `data-*` para um determinado plugin conforme mostrado a seguir:

```
$('body').off('dropdown.data-api'); // desabilita o plugin dropdown
```

Construtores jQuery estão disponíveis com a sintaxe clássica:

```
$.fn.nome-do-plugin.Constructor
```

5.2 Plugins

Nos itens a seguir estudaremos cada um dos plugins disponíveis no Bootstrap.

O script dos plugins do Bootstrap encontram-se no arquivo existente na pasta *js* denominado *bootstrap.min.js*.

No caso de download compilado esse arquivo contém *todos* os plugins do Bootstrap. Opcionalmente você poderá incluir somente os plugins que irá utilizará no seu projeto fazendo o download personalizado.

No caso de download de todo o Bootstrap (source) você poderá obter os arquivos dos plugins individualmente no arquivo *bootstrap.js* que encontra-se na pasta *js*.

5.2.1 Transições

O plugin para transições destina-se a ser usado em conjunto com outros plugins cujas funcionalidades admitem uso de transição tal como obtenção de efeitos slide e fade em janelas modais, abas, caixas de alerta e painéis. As transições disponíveis nesse plugin são `slide` e `fade`.

5.2.2 Janela modal

Janela modal é uma poderosa interface de diálogo com o usuário que se destina a variados usos e proporciona um visual agradável ao visitante da página.

O Bootstrap possibilita a criação de janela modal com uso de marcação típica e o mecanismo de abertura e fechamento da janela pode ser definido de duas formas: diretamente com uso do par atributo/valor `data-toggle="modal"` ou codificando com uso de métodos JavaScript previstos no Bootstrap. É possível ainda obter o efeito de janela modal criando scripts personalizados com uso dos métodos e propriedades próprios da biblioteca jQuery.

5.2.2.1 Com uso de data-toggle

Para acionar o mecanismo de abertura e fechamento da janela modal basta declarar o atributo `data-toggle` para o elemento que marca o botão ou o link de abertura da janela.

Os mecanismos internos do Bootstrap identificam aquele atributo na marcação e acionam os códigos JavaScript necessários para abrir e fechar a janela. O desenvolvedor não precisa criar código JavaScript para acionar a janela, eles são criados automaticamente pelo Bootstrap.

A marcação HTML mostrada a seguir é a padrão do Bootstrap para criação de uma janela modal com uso do atributo `data-toggle`.

- **HTML**

```
1. <button type="button" class="btn btn-primary btn-lg" data-toggle="modal"
       data-target="#modalLogin">
2.     Abrir janela modal
3. </button
4. <div  class="modal fade" id="modalLogin" tabindex="-1" role="dialog"
       aria-labelledby="modalLoginLabel">
5.     <div class="modal-dialog" role="document">
6.         <div class="modal-content">
7.             <div class="modal-header">>
8.                 <button type="button" class="close" data-dismiss="modal"
                   aria-label="Close"><span aria-hidden="true">&times;</span></button>
9.                 <h4 class="modal-title">Área administrativa</h4>
10.                </div>
11.                <div class="modal-body">
12                 <!-- conteúdo da janela modal -->
```

```
13.             </div>
14.                 <div class="modal-footer">
15.                     <button type="button" class="btn btn-default"
                            data-dismiss="modal">Fechar</button>
16.             </div>
17.         </div> <!-- /.modal-content -->
18.         </div> <!-- /.modal-dilog -->
19. </div> <!-- /.modal -->
```

[.../c5/modal-data-toggle.html]

Código comentado:

Linha(s)	Descrição
Linhas 1 a 3	Botão para abertura da janela modal. O atributo `data-target` deve apontar para o valor do atributo `id` do container geral da janela, no nosso caso o `div#modalLogin` na linha 4. Opcionalmente em lugar de um botão para abrir a janela poderíamos usar um link e neste caso o atributo `href` do link apontaria para o `id` do container geral da janela, no nosso caso `href="#modalLogin"`. A classe `btn` definida para o link de abertura da janela faz com que os mecanismos internos do Bootstrap transformem o link em um botão. Esse botão pode ser estilizado com uso de classes adicionais, tal como `btn-primary` e `btn-lg`. A classe `data-toggle`, definida para aquele link, informa ao Boostrap que a janela modal terá seu script de abertura e fechamento inserido automaticamente sem necessidade de codificação JavaScript pelo autor.
Linha 4 a 19	Marcação do `div` container da janela modal. A classe `modal` é de uso obrigatório e informa ao Boostrap que trata-se do container de uma janela modal. A classe `fade` faz com que o efeito de abertura seja por deslizamento da janela no sentido vertical, na direção de cima para baixo e com efeito de esmaecimento. Opcionalmente podemos omitir o valor `fade` que produz o efeito de mostrar e esconder a janela sem dislazamento e esmaecimento, ou seja, abertura e fechamento fixa na sua posição. Sugiro fazer uma cópia do exemplo retirar a classe `fade` verificando na prática os dois efeitos de abertura/fechamento. O atributo `id` é de livre escolha do autor.
Linhas 5 e 6	Containeres auxiliares para estilização e scripts.
Linhas 7 a 10	Marcação do `div.modal-header` container do cabeçalho da janela. Esse `div` contém o botão superior direito de fechamento da janela (um xis) e o título da janela. O atributo `data-dismiss` com o valor `modal` informa ao Bootstrap que se trata de um botão para fechar a janela. A classe `close` estiliza e posiciona o botão x(xis) de fechamento da janela.
Linha 11 a 13	Marcação do `div.modal-body` container do corpo (conteúdo principal) da janela.

Linha(s)	Descrição (cont.)
Linha 14 a 16	Marcação do `div.modal-footer` container do rodapé da janela. Esse `div` contém o botão inferior de fechamento da janela. O atributo `data-dismiss` com o valor `modal` informa ao Bootstrap que se trata de um botão para fechar a janela.

A renderização da janela modal constante do arquivo anterior é conforme mostrada na figura 5.1.

Figura 5.1 – Janela modal.

5.2.2.2 Com uso de JavaScript nativo

A segunda opção para acionar o mecanismo de abertura e fechamento da janela modal é com uso dos métodos JavaScript nativos do Bootstrap.

Na verdade, nas duas opções o mecanismo usa JavaScript a diferença é que para a primeira opção são os mecanismos internos do Boostrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para abrir e fechar a janela e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar a janela.

O método nativo para inicializar uma janela modal tem a sintaxe conforme mostrada a seguir.

```
$('#modalLogin').modal(options)
```

onde `#modalLogin` é o valor do `id` do container da janela modal. O nome desse valor, obviamente, é de livre escolha do autor e no nosso exemplo usamos `modalLogin`.

As opções para esse método são:

Nome	Tipo	Default	Descrição
backdrop	boleano	true	Define a existência de um fundo na cor preta e transparente para a janela modal. Pode-se ainda definir o valor static que define o fundo, mas desabilita o fechamento da janela quando clicamos nesse fundo.
keyboard	boleano	true	Define o fechamento da janela modal quando a tecla Esc é pressionada.
show	boleano	true	Mostra a janela modal na inicialização da página.
remote	path	false	O valor para essa opção é o url de um arquivo remoto cujo conteúdo será carregado com uso do método AJAX load da biblioteca jQuery e injetado no container .modal-content. Alternativamente podemos usar o atributo href na marcação HTML para definir o url remoto a carregar, conforme mostrado no exemplo a seguir. <a data-toggle="modal" href="remoto.html" data-target="#modalLogin"> Abrir janela modal Esta funcionalidade está em desuso e será retirada na versão BS4.

Opcionalmente, podemos definir as opções do método JavaScript nativo declarando-as com uso dos atributos `data-*`. Por exemplo: `data-keyboard="false"`.

São previstos ainda os métodos conforme as sintaxes mostradas a seguir.

```
$('#modalLogin').modal("toggle")  // Abre e fecha a janela modal
$('#modalLogin').modal("show")    // Abre a janela modal
$('#modalLogin').modal("hide")    // Fecha a janela modal
```

Os eventos nativos são descritos a seguir.

Evento	Descrição
`show.bs.modal`	Esse evento ocorre imediatamente após a chamada de uma instância do método show.
`shown.bs.modal`	Esse evento ocorre assim que a janela modal torna-se visível para o usuário (após as transições CSS terem ocorrido).
`hide.bs.modal`	Esse evento ocorre imediatamente após a chamada de uma instância do método hide.
`hidden.bs.modal`	Esse evento ocorre assim que a janela modal torna-se invisível para o usuário (após as transições CSS terem ocorrido).
`loaded.bs.modal`	Esse evento ocorre assim que um conteúdo remoto requisitado é carregado na janela modal.

No exemplo a seguir mostramos o uso de métodos JavaScript para abertura de janela modal e a captura de eventos para abrir uma caixa de alerta JavaScript identificando o evento disparado. A marcação HTML é a mesma marcação usada no anterior com a única diferença que dela retiramos o par atributo valor `data-toggle="modal"` conforme mostrado a seguir.

- **HTML**

```
1. <button type="button" class="btn btn-primary btn-lg" data-toggle="modal" data-
   target="#modalLogin">
2.    Abrir janela modal
3. </button
4. <div  class="modal fade" id="modalLogin" tabindex="-1" role="dialog" aria-
   labelledby="modalLoginLabel">
5.    <div class="modal-dialog" role="document">
...<!-- idêntico ao exemplo anterior -->
18.       </div> <!-- /.modal-dilog -->
19. </div> <!-- /.modal -->
```

Feitas essa única alteração na marcação HTML criamos o script conforme mostrado a seguir.

- **JavaScript**

```
<script type="text/javascript">
   $('button[data-target="#modalLogin"]').on('click', function() {
      $('#modalLogin').modal('toggle');
   });
$('#modalLogin').on('show.bs.modal', function () {
   alert('Modal vai ser aberta\nEvento: show.bs.modal');
});
$('#modalLogin').on('shown.bs.modal', function () {
   alert('Modal foi aberta\nEvento: shown.bs.modal');
});
$('#modalLogin').on('hide.bs.modal', function () {
   alert('Modal vai ser fechada\nEvento:hide.bs.modal');
});
$('#modalLogin').on('hidden.bs.modal', function () {
   alert('Modal foi fechada\nEvento:hidden.bs.modal');
});
</script>
</body>
</html>
```

[.../c5/modal-javascript.html]

Visualize esse exemplo online para constatar o disparo dos eventos `show`, `shown`, `hide` e, `hidden`.

5.2.3 Dropdown

Dropdown é uma funcionalidade que permite ao autor criar um efeito de abertura e fechamento de submenus. Podemos inserir dropdown nos componentes barras de navegação, abas e pills estudados no capítulo 4.

O Boostrap possibilita a criação de menu dropdown com uso de marcação típica e o mecanismo de abertura do menu pode ser definido de duas formas: diretamente com uso do par atributo/valor `data-toggle="dropdown"` ou codificando com uso do método JavaScript `dropdown()` previsto no Bootstrap.

5.2.3.1 Com uso de data-toggle

A marcação HTML mostrada a seguir é a padrão do Bootstrap para criação de um menu dropdown criado com uso do par atributo/valor `data-toggle="dropdown"` e inserido em uma barra de navegação (ver [4.7]).

Notar que o valor do atributo `data-target` do botão contido em `div.navbar-header` deve ser igual ao `id` do container geral do menu dropdown.

- **HTML**

```
<nav class="navbar navbar-default">
   <div class="navbar-header">
   <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
      data-target="#barraDataToggle">
      ...
      <a class="navbar-brand" href="#">Automóveis</a>
   </div>
   <div class="collapse navbar-collapse" id="barraDataToggle">
      <ul class="nav navbar-nav">
         <li class="active"><a href="#">Home <span class="sr-only">(current)</span></a></li>
         <li><a href="#">Ford</a></li>
         <li class="dropdown">
               <a href="#" class="dropdown-toggle" data-toggle="dropdown"
               role="button" aria-expanded="false">Renault<span class="caret"></span></a>
            <ul class="dropdown-menu" role="menu">
               <li><a href="#">Clio</a></li>
               ...
            </ul>
         </li>
         <li class="dropdown">
```

```
                <a href="#" class="dropdown-toggle" data-toggle="dropdown"
                role="button" aria-expanded="false">Volkswagen<span class="caret"></span></a>
              <ul class="dropdown-menu" role="menu">
                <li><a href="#">Novo Gol</a></li>
                ...
              </ul>
            </li>
            <li><a href="#">Contato</a></li>
          </ul><!-- /.navbar -->
        </div><!-- /.navbar-collapse -->
      </nav>
```

[.../c5/dropdown-data-toggle.html]

A renderização da barra de navegação com link com efeito dropdown conforme mostada na marcação HTML anterior é conforme a figura 5.2.
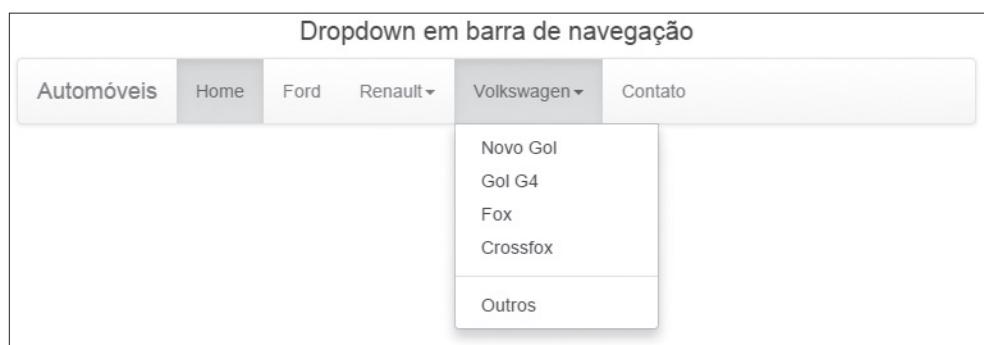
Figura 5.2 – Dropdown em barra de navegação.

5.2.3.2 Com uso de JavaScript nativo

A segunda opção para obter o efeito dropdown é com uso dos métodos JavaScript nativos do Bootstrap.

Na verdade, nas duas opções o mecanismo usa JavaScript a diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para fazer funcionar o dropdown e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar o dropdown.

O método nativo para inicializar um dropdown tem a sintaxe conforme mostrada a seguir.

```
$('#dropDownN').dropdown()
```

onde `#doprDownN` é o valor do `id` do elemento `ul` que contém os liks do dropdown. O nome desse valor, obviamente, é de livre escolha do autor e no nosso exemplo usamos `dropDown1` para o submenu do link "Renault" e `dropDown2` para o submenu do link "Volkswagen" pois mostramos estes dois links com dropdown na bara de navegação.

Este método não admite opções.

Os eventos nativos são descritos a seguir.

Evento	Descrição
`show.bs.dropdown`	Esse evento ocorre imediatamente após a chamada de uma instância do método show.
`shown.bs.dropdown`	Esse evento ocorre assim que a janela modal torna-se visível para o usuário (após as transições CSS terem ocorrido).
`hide.bs.dropdown`	Esse evento ocorre imediatamente após a chamada de uma instância do método hide.
`hidden.bs.dropdown`	Esse evento ocorre assim que a janela modal torna-se invisível para o usuário (após as transições CSS terem ocorrido).

Estes eventos são atrelados aos elementos que marcam os links de abertura do dropdown e no nosso exemplo adotamos o `id="abreDropDown1"` para o link "Renault" e `id="abreDropDown2"` para o link "Volkswagen".

Se os eventos `show.bs.dropdown` e `shown.bs.dropdown` forem disparados por ação de `click`, está disponível a propriedade `relatedTarget` que retorna o elemento que foi clicado.

No exemplo a seguir mostramos o uso do método JavaScript para abertura de dropdown, para a captura de eventos e para inspecionar a propriedade `relatedTarget`. A marcação HTML é a mesma marcação usada no exemplo anterior, com acréscimo dos atributos `id` conforme mostrado em destaque na marcação HTML a seguir.

Alerta: Ao contrário do componente modal estudado anteriormente para o componente dropdown é necessário que o par atributo valor `data-toggle="dropdown"` *não seja retirado da marcação* HTML.

- **HTML**

```
/* igual à marcação do exemplo anterior */
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
      aria-expanded="false">
```

```
      Renault<span class="caret" id="abreDropDown1"></span></a>
         <ul class="dropdown-menu" role="menu" id="dropDown1">
            <li><a href="#">Clio</a></li>
            ...
         </ul>
   </li>
      <li class="dropdown">
         <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
            aria-expanded="false">
         Volkswagen<span class="caret" id="abreDropDown2"></span></a>
            <ul class="dropdown-menu" role="menu" id="dropDown2">
               <li><a href="#">Novo Gol</a></li>
               ...
            </ul>
      </li>
      <li><a href="#">Contato</a></li>
      </ul><!-- /.navbar -->
   </div><!-- /.navbar-collapse -->
</nav>
```

O script é conforme mostrado a seguir.

- **JavaScript**

```
<script type="text/javascript">
   $('#abreDropDown1, #abreDropDown2"]').on('click', function() {
      $('#dropDown1, #dropDown2').dropdown('toggle');
   });
$('.dropdown').on('show.bs.dropdown', function (e) {
   var textoLink = $(e.relatedTarget).text();
   alert('Dropdown vai ser aberto\nEvento: show.bs.dropdown\nLink clicado: ' +
textoLink);
});
$('.dropdown').on('shown.bs.dropdown', function () {
   alert('Dropdown foi aberto\nEvento: shown.bs.dropdown');
});
$('.dropdown').on('hide.bs.dropdown', function () {
   alert('Dropdown vai ser fechado\nEvento: hide.bs.dropdown');
});
$('.dropdown').on('hidden.bs.dropdown', function () {
   alert('Dropdown foi fechado\nEvento: hidden.bs.dropdown');
});
```

```
</script>
</body>
</html>
```

[.../c5/dropdown-javascript.html]

Visualize esse exemplo online para constatar o disparo dos eventos `show`, `shown`, `hide` e, `hidden` e o uso da propriedade `relatedTarget`.

5.2.4 Scrollspy

Scrollspy é uma funcionalidade que permite ao autor monitorar automaticamente a posição da barra de rolagem da página quando se tem navegação interna (in-page), ou seja, navegação na própria página.

Navegar na própria página significa que o clique em link de um mecanismo de navegação em lugar de conduzir o usuário para uma nova página (página dentro do mesmo site ou externa) é conduzido a um determinado ponto da própria página (aquela que contém o mecanismo de navegação).

Em geral navegação interna é indicada para páginas extensas com seções que podem ser consultadas (lidas) individualmente sem necessidade de leitura corrida para seu entendimento. Por exemplo: página FAQ desenvolvida sem o recurso do menu sanfona, na qual as perguntas estão inseridas como itens de um menu e as respostas como alvo do clique em uma pergunta.

Outra forma de navegação interna, para a qual podemos aplicar a funcionalidade scrollspy é aquela na qual os conteúdos alvo dos links são inseridos dentro de um container com altura fixa e existência de barra de rolagem para acessar conteúdos em *overflow* com aquela altura.

O mecanismo do Bootstrap para fazer funcionar o scrollspy consiste em destacar o link quando o usuário chega ao seu alvo por ação de rolagem na barra de rolagem ou quando o usuário aciona a barra de rolagem atingindo os alvos dos links. Só faz sentido projetar scrollspy quando o menu de navegação permanece fixo e à vista do usuário durante a rolagem.

A marcação HTML típica para navegação interna é conforme mostrada a seguir.

- **HTML**

```
<a href="#css">CSS</a>
<a href="#html">HTML</a>
<a href="#js">javascript</a>
...
<div id="css">CSS</div>
   <!-- texto CSS -->
<div id="css">HTML</div>
   <!-- texto HTML -->
<div id="css">javascript</div>
   <!-- texto javascript -->
...
```

O atributo `href` de cada link é igual ao valor do atributo `id` precedido do sinal # (tralha) definido para o elemento da página para o qual se pretende navegar. Na marcação anterior um elemento `div`.

Existem duas maneiras de se projetar scrollspy: diretamente com uso do par atributo/valor `data-spy="scroll"` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

5.2.4.1 Scrollspy para a página com uso de data-spy

Para acionar o mecanismo scrollspy basta declarar os atributos `data-spy` e `data-target` para o elemento cuja barra de rolagem se pretende monitorar. No nosso exemplo vamos monitorar a barra de rolagem vertical da página (janela do navegador).

Os mecanismos internos do Bootstrap identificam aqueles atributos na marcação e acionam os códigos JavaScript necessários para fazer funcionar o scrollspy. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

No exemplo a seguir mostramos os códigos para a criação de uma barra de navegação fixa (ver [4.7.2]) no topo da página destinada à navegação interna e utilizamos a funcionalidade scrollspy para monitorar a barra de rolagem vertical da página.

- **HTML**

```
1. <body data-spy="scroll" data-target="#barraNavegacao">
2. <div class="container-fluid" id="home">
3. ...
4. <nav class="navbar navbar-default navbar-fixed-top" id="barraNavegacao">
```

```
5.  <div class="container-fluid">
6.  div class="navbar-header">
7.          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
               data-target="#barraFixaTop">
8.          ...
9.          </button>
10. <a class="navbar-brand" href="#">Site</a>
11.             </div>
12. <div class="collapse navbar-collapse" id="barraFixaTop">
13. <ul class="nav navbar-nav">
14.          <li class="active"><a href="#home">Home<span class="sr-only">(current)</span>
               </a></li>
15.         <li><a href="#portfolio">Portfólio</a></li>
16.         <li><a href="#materias">Matérias</a></li>
17.         <li><a href="#contato">Contato</a></li>
18.     </ul>
19  </div><!-- /.navbar-collapse -->
20. </div><!-- /.container-fluid -->
21. </nav>
22. <div id="portfolio"> ... </div>
23. <div id="materias">... </div>
24. <div id="contato"> ... </div>
25. ...
```

[.../c5/scrollspy-body-data-spy.html]

Código comentado:

Linha(s)	Descrição
Linha 1	Para fazer funcionar o plugin scrollspy basta acrescentar os pares atributo/valor `data-spy="scroll"` e `data-target="valor"` no elemento `body` da página a monitorar a rolagem. O valor do atributo `data-target` é o do seletor que define o container do mecanismo de navegação, no nosso exemplo `nav#barraNavegacao` que é o `id` do elemento `nav` container da barra de navegação.
Linhas 4 a 21	Marcação que cria uma barra de navegação fixa no topo da página. Ver [4.7.2]. Notar na linha 4 o container da barra de navegação com o `id` de valor igual `barraNavegacao` que foi referenciado no atributo `data-target` do elemento `body` como descrito no item anterior.
Linhas 14 a 17	Marcação dos links da barra de navegação com o atributo `href` apontando para o `id` do container de conteúdos de cada link como mostrado nas linhas 2, 22, 23 e 24 respectivamente.

Em [4.7.2], quando estudamos barras de navegação fixas dissemos o seguinte:

> Ao optar pelo uso de uma barra de navegação fixa é preciso definir um padding de no mínimo 50px para o elemento body com a finalidade de abrir espaço para inserção da barra. Declare padding-top e/ou padding-bottom conforme use barra fixa no topo e/ou rodapé.

Nesse exemplo, ao clicarmos um link, a página rola para o texto do elemento alvo do link (um elemento `div`) que é colocado no topo da janela. Como temos uma barra de navegação fixa no topo ela irá cobrir o início daquele texto até uma altura de 50px, que é a altura da barra de navegação. Então, precisamos declarar regras de estilo para fazer o texto alvo do link descer no mínimo 50px e sair de trás da barra de navegação. Para isso acrescente a seguinte folha de estilo à página:

- **CSS**

```
body { padding-top:50px;background: #ccc; }
#portfolio, #materias, #contato {
   padding-top: 50px;
/* O padding-top aqui é necessário para posicionar o container do conteúdo abaixo da
barra fixa */
}
```

Os valores definidos na folha de estilo mostrada são válidos para a aplicação padrão de estilos do Bootstrap em layouts fixos. Se necessário altere aqueles valores para contemplar as particularidades do seu projeto.

Ao consultar esse arquivo clique os links da barra de navegação e depois movimente a barra de rolagem da página para cima e para baixo observando o plugin scrollspy em ação.

5.2.4.2 Scrollspy em container com uso de data-spy

No exemplo a seguir mostraremos a criação de uma barra de navegação básica destinada à comandar a navegação em um container da página e utilizamos a funcionalidade scrollspy para a barra de rolagem vertical do container. A barra de rolagem no container será criada com uso das propriedades CSS `height` e `overflow` para o container.

Usaremos a mesma marcação HTML do exemplo anterior com as modificações conforme mostradas em destaque a seguir.

- **HTML**

```
1. <body data-spy="scroll" data-target="#barraNavegacao">
2. <div class="container-fluid" id="home">
3. ...
4. <nav class="navbar navbar-default navbar-fixed-top" id="barraNavegacao">
5. <div class="container-fluid">
6. div class="navbar-header">
7.     <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
           data-target="#barraFixaTop">
8.      ...
9.     </button>
10.     <a class="navbar-brand" href="#">Site</a>
11.            </div>
12. <div class="collapse navbar-collapse" id="barraFixaTop">
13.    <ul class="nav navbar-nav">
14.         <li class="active"><a href="#home">Home<span class="sr-only">(current)</span></a></li>
15.        <li class="active"><a href="#portfolio">Portfólio<span class="sr-
    only">(current)</span></a></li>
16.        <li><a href="#materias">Matérias</a></li>
17.        <li><a href="#contato">Contato</a></li>
18.     </ul>
19 </div><!-- /.navbar-collapse -->
20. </div><!-- /.container-fluid -->
21. </nav>
22. <div id="containerConteudo" data-spy="scroll" data-target="#barraNavegacao">
23.    <div id="portfolio"> ... </div>
24.    <div id="materias">... </div>
25.    <div id="contato"> ... </div>
26 </div>
27. ...
```

[.../c5/scrollspy-container.html]

Nesse exemplo forçamento o aparecimento de uma barra de rolagem no container. Para isso acrescente a seguinte folha de estilo à página:

- **CSS**

```
<style type="text/css">
#containerConteudo {
   height: 400px;
```

```
    overflow-y: scroll;
  }
  </style>
```

Ao consultar esse arquiro clique os links da barra de navegação e depois movimente a barra de rolagem do container para cima e para baixo observando o plugin scrollspy em ação.

5.2.4.3 Scrollspy com JavaScript nativo

Na verdade, nas opções com uso de `data-spy` e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para fazer funcionar scrollspy e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar a janela.

O método nativo para inicializar scrollspy tem a sintaxe conforme mostrada a seguir.

```
$('xpto').scrollspy(options)
```

onde `xpto` identifica o container do elemento a monitorar a barra de rolagem. O escolha desse identificador, obviamente, é de livre escolha do autor.

Existe uma opção para esse método conforme descrita a seguir.

Nome	Tipo	Default	Descrição
offset	número	10	Retorna o valor em pixel da distância até o topo quando efetua-se o cálculo da posição da barra de rolagem. Importante notar que essa opção não define a distância, ela apenas retorna a distância.

É previstos ainda o método conforme a sintaxe mostrada a seguir.

```
$('xpto').spy('refresh') // Para inserções dinâmicas no DOM
```

Esse método é útil para atualizar o plugin quando há inserção dinâmica no DOM já carregado. A sintaxe geral para aplicar o método em todos os scrollspy constantes da página é mostrada a seguir.

```
$('[data-spy="scroll"]').each(function () {
  var $spy = $(this).scrollspy('refresh')
});
```

O evento nativo é descrito a seguir.

Evento	Descrição
activate	Esse evento ocorre quando um novo link é ativado pelo scrollspy.

A seguir mostraremos o mesmo exemplo constante do item [5.2.4.1] anterior para demonstrar o uso de JavaScript nativo para essa funcionalidade.

- **HTML**

```
<body> //retiramos os atributos data-*
<!-- igual à marcação do item [5.2.4.1] -->
```

[.../c5/scrollspy-javascript.html]

Notar que foram retirados os atributos `data-*` do elemento `body`.

O código JavaScript é mostrado a seguir.

- **JavaScript**

```
<script type="text/javascript">
   $('body').scrollspy( {target: '#barraNavegacao'} ).on('activate.bs.scrollspy',
      function () {
         alert('Foi ativado scrollspy');
      });
</script>
</body>
</html>
```

Ao consultar esse arquivo clique os links da barra de navegação e depois movimente a barra de rolagem da página para cima e para baixo observando o plugin scrollspy em ação. Observe o disparo do evento `activate.bs.scrollspy` toda vez que se clica um link na caixa de alerta JavaScript.

5.2.5 Abas

Abas é uma funcionalidade que permite ao autor criar um mecanismo de navegação baseado em abas, ou seja, quando o usuário clica em um link contido em uma aba (ou em pill) o conteúdo alvo do link é apresentado em uma área fixa na página.

Existem duas maneiras de se projetar navegação em abas: diretamente com uso do par atributo/valor `data-toggle="tab"` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

A criação de abas e pills foi estudada em [4.6] onde mostramos como criar os diferentes tipos de aba e pill, mas sem acrescentar a funcionalidade da navegação. Veremos a seguir abas em funcionamento.

5.2.5.1 Abas com uso de data-toggle

Para acionar o mecanismo de navegação em abas basta declarar os atributos `data-toggle="tab"` para cada um dos elementos `a` que marcam os links das abas.

Os mecanismos internos do Bootstrap identificam aquele atributo na marcação e acionam os códigos JavaScript necessários para fazer funcionar a navegação em aba. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

No exemplo a seguir mostraremos a criação de uma aba de navegação padrão.

- **HTML**

```
1.  <body>
2.  <ul class="nav nav-tabs" role="tablist">
3.      <li role="presentation" class="active"><a href="#home" aria-controls="home"
            role="tab" data-toggle="tab">Home</a></li>
4.      <li role="presentation"><a href="#portfolio" aria-controls="portfolio" role="tab"
             data-toggle="tab">Portfólio</a></li>
5.      <li role="presentation"><a href="#artigos" aria-controls="artigos" role="tab"
            data-toggle="tab">Artigos</a></li>
6.      <li role="presentation"><a href="#contato" aria-controls="contato" role="tab"
            data-toggle="tab">Contato</a></li>
7.  </ul>
8.  <div class="tab-content">
9.      <div role="tabpanel" class="tab-pane fade in active" id="home">
11.         <h4>Home</h4>
12.         <p>Lorem ipsum...</p>
13.     </div>
14.     <div role="tabpanel" class="tab-pane fade" id="portfolio">
15.         <h4>Portfólio</h4>
16.         <p>Curabitur hendrerit...</p>
17.     </div>
19.     <div role="tabpanel" class="tab-pane fade" id="artigos">
20.         <h4>Artigos</h4>
21.             <p>Pellentesque sapien...</p>
22.     </div>
```

```
19.     <div role="tabpanel" class="tab-pane fade" id="contato">
20.         <h4>Contato</h4>
21.         <p>Cum sociis natoque...</p>
22.     </div>
23. </div> <!-- /.tab-content -->
```

[.../c5/aba-data-toggle.html]

Código comentado:

Linha(s)	Descrição
Linhas 2 a 7	Links da aba. O atributo `href` do link tem como alvo o valor do `id` do container dos conteúdos de cada aba.
Linhas 8 e 23	Container de todo o conteúdo a ser exibido nas abas. Esse container deverá receber a classe `tab-content`.
Linhas 9 a 22	Cada um dos conteúdos de uma aba deverá estar dentro de um container `div` ao qual atribui-se a classe `tab-pane` e o `id` com valor igual ao valor do atributo `href` (sem o sinal tralha) do link que mostra o conteúdo. Notar as classes `fade` e `in` no container, elas se destinam a adicionar o efeito fade na abertura/fechamento das abas e a classe `in` deve ser usada somente na aba inicialmente ativa.

Dica: Os atributos `role` e `aria-*` constantes da marcação são usados para contemplar critérios de acessibilidade.

Neste exemplo criamos uma folha de estilos incorporada ao arquivo com a finalidade de personalizar as abas. A folha de estilos é mostrada e comentada a seguir.

- **CSS**

```
<style rel="stylesheet">
   body { background: #ccc;}
   h2 {font-size:20px;}
      .tab-content {
      height: 250px;   /* limitamos a altura da ára de mostra dos conteúdos */
      overflow-y:auto;  /* barra de rolagem para conteúdos que ocupem altura
                           maior que o limite 250px */
      border: 3px solid #fff;
      border-bottom-left-radius:5px;
      border-bottom-right-radius:5px;
   }
   .tab-content h2 { margin-top:3px; }     /* sobrescrevemos a margem padrão de h2 */
   #home,#portfolio,#artigos,#contato {
```

```
        color: #fff;
        padding: 0 15px;
    }
    #home { background: #333; }
    #portfolio { background: #930;}
    #artigos { background: #030;}
    #contato { background: #036;}
    .nav-tabs > li > a {     /* crimos cor de fundo para os links nas abas */
        background: #e0dbdb;
    }
</style>
```

5.2.5.2 Abas com JavaScript nativo

Na verdade, nas opções com uso de `data-toggle` e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para fazer funcionar a navegação em abas e pills e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar as abas.

O método nativo para inicializar abas tem a sintaxe conforme mostrada a seguir.

```
$('a').tab('show')
```

onde `a` identifica o elemento link na navegação da aba. Esses é o único método deste plugin e se destina a mostrar e esconder os conteúdos da abas.

Os eventos nativos são descritos a seguir.

Evento	Descrição
`show.bs.tab`	Esse evento ocorre quando o usuário clica uma aba, mas o conteúdo dela ainda não foi mostrado.
`shown.bs.tab`	Esse evento ocorre quando o conteúdo da aba clicada é mostrado.
`hide.bs.tab`	Esse evento ocorre antes de uma aba ser fechada em consequência de o usuário ter clicado em outra aba.
`hidden.bs.tab`	Esse evento ocorre depois que uma aba foi fechada em consequência de o usuário ter clicado em outra aba.

Em todos estes eventos estão disponíveis as propriedades `event.target` e `event.relatedTarget` que retornam o URL da aba a ser aberta e o URL da aba ativa (a ser fechada).

No exemplo a seguir usamos a mesma marcação do exemplo mostrado no item anterior [5.2.5.1], retirando apenas o atributo `data-toggle` da marcação HTML conforme mostrado a seguir.

- **HTML**

```
1.  <body>
2.  <ul class="nav nav-tabs" role="tablist">
3.      <li role="presentation" class="active"><a href="#home" aria-controls="home"
            role="tab" data-toggle="tab">Home</a></li>
4.      <li role="presentation"><a href="#portfolio" aria-controls="portfolio" role="tab"
            data-toggle="tab">Portfólio</a></li>
5.      <li role="presentation"><a href="#artigos" aria-controls="artigos" role="tab"
            data-toggle="tab">Artigos</a></li>
6.      <li role="presentation"><a href="#contato" aria-controls="contato" role="tab"
             data-toggle="tab">Contato</a></li>
7.  </ul>
8. a 22.  <!-- igual ao exemplo anterior -->
23.   </div> <!-- igual à marcação do item [5.2.5.1] -->
```

[.../c5/abas-javascript.html]

O código JavaScript é mostrado a seguir.

- **JavaScript**

```
...
<script type="text/javascript">
   $('a[role="tab"]').on('click', function(e) {
      e.preventDefault;
      $(this).tab('show');
   });
   $('a[role="tab"]').on('show.bs.tab', function (e) {
      var tabAbrir = e.target // Abra a ser aberta
      var tabFechar =  e.relatedTarget // Aba ativa a ser fechada
      alert('Aba: '+tabAbrir+ ' vai ser aberta\n\nAba: '+tabFechar+ ' vai ser fechada\n\
nEvento: show.bs.tab');
});
</script>
</body>
</html>
```

Ao consultar esse arquivo clique os links das abas. Observe a caixa de alerta JavaScript a cada clique em uma aba.

5.2.6 Tooltip

No jargão do desenvolvimento web a palavra inglesa tooltip foi traduzida para o português como *dica de contexto* É aquela nossa conhecida funcionalidade que permite abrir uma pequena janela popup, quando o usuário passa o mouse sobre uma palavra, contendo um texto explicativo, ou uma dica sobre a palavra.

Existem duas maneiras de se projetar tooltip: diretamente com uso do par atributo/valor `data-toggle="tooltip"` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

5.2.6.1 Tooltip com uso de data-toggle

Para criar um tooltip basta declarar o atributo `data-toggle="tooltip"` para o elemento HTML que deverá ser o container da palavra que irá disparar o tooltip. O texto da tooltip será o valor do atributo `title` daquele elemento. Observe a seguir a marcação típica para criar um tooltip em uma palavra de um texto corrente e em um botão.

```
<a href="#" data-toggle="tooltip" title="Fruta tropical">Pupunha</a>
<button type="button" class="btn btn-default" data-toggle="tooltip"
   title="Fruta tropical">Pupunha</button>
```

Os mecanismos internos do Bootstrap identificam aquele atributo na marcação e acionam os códigos JavaScript necessários para fazer funcionar o tooltip. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

Por padrão o texto do tooltip é na cor branca, inserido em uma caixa na cor preta com bordas arredondadas e posicionado na parte superior da palavra que dispara o tooltip.

É possível alterar a posição da caixa com uso do atributo `data-placement` e seus valores: `top`, `right`, `bottom` e `left`, sendo `top` o valor padrão. Mostramos a seguir as sintaxes para posicionamento.

```
// padrão no topo
<a href="#" data-toggle="tooltip" title="Fruta tropical">Pupunha</a>
// à direita
<a href="#" data-toggle="tooltip" data-placement="right" title="Fruta tropical">Pupunha</a>
// embaixo
<a href="#" data-toggle="tooltip" data-placement="bottom" title="Fruta tropical">Pupunha</a>
// à esquerda
<a href="#" data-toggle="tooltip" data-placement="left" title="Fruta tropical">Pupunha</a>
```

No exemplo a seguir mostraremos a criação de tooltips nas quatro posições possíveis.

- **HTML**

```
<p>As frutas de
<a href="#" data-toggle="tooltip" title="média acima de 10°C...">clima temperado</a>
são ... lenhosos ou
<a href="#" data-toggle="tooltip"  data-placement="left"
   title="plantas trepadeiras lenhosas">lianas</a>.
Contudo elas não ...
As principais frutas de clima temperado são a
<a href="#" data-toggle="tooltip" data-placement="right" title="É um fruto ...">ameixa</a>,
a ameixa-japonesa, a cereja-doce, o damasco, a
<a href="#" data-toggle="tooltip" data-placement="bottom"
   title="Nome científico: Rubus idaeus">framboesa</a>, a maçã....</p>
/* Mais quatro exemplos com tooltips em botões */
```

- **JavaScript**

```
<script type="text/javascript">
   $('[data-toggle="tooltip"]').tooltip();
</script>
</body>
</html>
```

[.../c5/tooltip-alinhamento.html]

O componente tooltip, mesmo quando criado com uso do atributo `data-toggle` precisa, obrigatoriamente, ser inicializado manualmente pelo desenvolvedor, conforme o código JavaScript mostrado, ou seja, deverá ser declarado o método `tooltip()` para o elemento que o dispara (no nosso exemplo `[data-toggle="tooltip"]`).

Ao visualizar este exemplo passe o mouse sobre as palavras em destaque (cor azul por padrão) no texto e sobre os botões para ver o tooltip em ação nas suas quatro posições possíveis.

5.2.6.2 Tooltip com JavaScript nativo

Na verdade, nas opções com uso de `data-toggle` e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação

HTML e acionam os códigos JavaScript necessários para fazer funcionar o tooltip e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar o tooltip.

O método nativo para inicializar um tooltip tem a sintaxe conforme mostrada a seguir.

```
$('seletor').tooltip(options)
```

onde `seletor` é uma referência ao elemento container da palavra que dispara o tooltip.

As opções para esse método são:

Nome	Tipo	Default	Descrição
animation	boleano	true	Aplica o efeito fade ao mostrar/esconder o tooltip.
html	boleano	false	O valor marcação HTML define que o conteúdo do tooltip será a marcação HTML aqui definida em lugar de texto simples.
placement	string ou função	'top'	Define o posicionamento do tooltip em relação à palavra que o dispara. Os valores possíveis são: top, right, bottom e left. Admite ainda a combinação de auto com um dos quatro posicionamentos, por exemplo: 'auto left' que posiciona à esquerda quando há espaço à esquerda e à direita caso contrário.
selector	string	false	Essa opção permite que se delegue a função de disparar o tooltip a um determinado seletor. Isso faz com que seletores inseridos no DOM dinamicamente herdem a funcionalidade tooltip.
title	string ou função	''	Com uso dessa opção podemos definir um texto padrão para o tooltip. Para visualizar o texto padrão é necessário que o atributo title não seja definido no elemento container do tooltip. Se for definido, sobrescreverá o texto padrão.
trigger	string	'hover focus'	O valor para essa opção define o evento que dispara o tooltip. Os valores possíveis são: click, hover, focus e manual. É admitido usar mais de um valor e nesse caso devem-se separar os valores com espaço. O valor manual não pode ser combinado com outro valor.
delay	número ou object	0	Define o tempo, em milissegundos, de retardo para abertura e fechamento do tooltip. Por exemplo, número: delay: 200 ou objeto: delay: { show: 100, hide: 200 }. Não se aplica para a opção trigger: “manual”.
container	string ou false	false	Define um container ao qual o tooltip será anexado. Por exemplo: container: '#janelaModal'.

Nome	Tipo	Default	Descrição
template	string	'<div class=”tooltip” role=”tooltip”> <div class=”tooltip-arrow”> </div> <div class=”tooltip-inner”> </div></div>'	Marcação HTML base a usar para criar um tooltip. O valor do atributo title será injetado no elemento com a classe tooltip-inner, a setinha será injetada no elemento com a classe tooltip-arrow. O container geral é no elemento com a classe .tooltip.
viewport	string ou object ou function	{ selector: 'body', padding: 0 }	Confina o tooltip aos limites do seu elemento.

São previstos ainda os métodos conforme as sintaxes mostradas a seguir.

```
$('seletor').tooltip( "show" )    // Mostra o tooltip
$('seletor').tooltip( "hide" )    // Esconde o tooltip
$('seletor').tooltip( "toggle" )  // Alterna mostra/esconde o tooltip
$('seletor').tooltip( "destroy")  // Esconde e destrói o tooltip
```

Os eventos nativos são descritos a seguir.

Evento	Descrição
show.bs.tooltip	Esse evento ocorre quando o usuário dispara o tooltip, mas o conteúdo dele ainda não foi mostrado.
shown.bs.tooltip	Esse evento ocorre quando o conteúdo do tooltip é mostrado.
hide.bs. tooltip	Esse evento ocorre imediatamente antes de o tooltip ser fechado.
hidden.bs. tooltip	Esse evento ocorre depois que o tooltip foi fechado.
inserted.bs.tooltip	Esse evento ocorre depois do evento show.bs.tooltip quando o template do tooltip for adicionado ao DOM.

O exemplo a seguir demonstra a criação de tooltips acionados por JavaScript nativo, bem como o uso de algumas opções do método `tooltip()`.

- **HTML**

```
<div class="row">
<div class="col-md-6 um">
   <h4>Opções:<br>
      ...
```

```
        <p>A <a href="#">maçã</a> é uma das principais frutas do clima temperado.</p>
</div>
<div class=" col-md-6 dois">
    <h4>Opções:<br>
        ...
        <p>A <i>maçã</i> é uma das principais frutas do clima temperado.</p>
</div>
</div> <!-- /.row -->
<div class="row">
<div class=" col-md-6 tres">
    <h4>Opções:<br>
        ...
        <p>A <a href="#">maçã</a> é uma das principais frutas do clima temperado.</p>
</div
<div class=" col-md-6 quatro">
    <h4>Opções:<br>
        ...
        <p>A <a href="#">maçã</a> é uma das principais frutas do clima temperado.</p>
</div>
</div> <!-- /.row -->
<div class="row">
    <div class=" col-lg-12 ">
        <p><a href="#" class="cinco">Clique aqui</a> para mostrar/esconder o segundo
tooltip da página.</p>
    </div> <!-- /.col-lg-12 -->
</div> <!-- /.row -->
```

[.../c5/tooltip-javascript.html]

Notar que não há necessidade de se usar atributos `data-*`.

O código JavaScript é mostrado a seguir.

- **JavaScript**

```
...
<script type="text/javascript">
$('.um a').tooltip( {
    trigger: 'click',
```

```
    title: 'Contém vitaminas B1 e B12',
    placement: 'right'
});

$('.dois i').tooltip( {
    html: true,
    title: '<img src="maca.jpg">',
    placement: 'top'
});

$('.tres a').tooltip( {
    html: true,
    title: '<h3>Título do texto</h3><p>Parágrafo com o texto</p>',
    delay: { show: 500, hide: 1000 }
});

$('.quatro a').tooltip( {
    html: true,
    title: '<img src="maca.jpg">',
    animation: false
});

$('.cinco').on('click', function(event) {
    event.preventDefault();
    $('.dois i').tooltip('toggle');
});
</script>
</body>
</html>
```

Ao consultar esse arquivo passe o mouse (ou clique) a palavra maçã em cada um dos quatro textos para verificar o disparo do tooltip.

5.2.7 Popover

Popover é uma funcionalidade semelhante ao tooltip. Destina-se a fornecer informação adicional sobre uma determina palavra de um texto. Enquanto o tooltip é indicado para apresentar uma pequena dica informativa o popover destina-se a fornecer uma explicação mais detalhada.

Existem duas maneiras de se projetar popover: diretamente com uso do par atributo/valor `data-toggle="popover"` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

5.2.7.1 Popover com uso de data-toggle

Para criar um popover basta declarar o atributo `data-toggle="popover"` para o elemento HTML que deverá ser o container da palavra que irá disparar o popover. O título do popover será o valor do atributo `title` e o texto do popover será o valor do atributo `data-content` daquele elemento. Observe a seguir a marcação típica para criar um popover em uma palavra de um texto corrente e em um botão.

```
<a href="#" data-toggle="popover" title="Informações nutricionais"
   data-content="O consumo...">maçã</a><button type="button"
   class="btn btn-default" data-toggle="popover"
   title="Informações nutricionais" data-content="O consumo...">maçã </button>
```

Os mecanismos internos do Bootstrap identificam os atributos `data-*` na marcação e acionam os códigos JavaScript necessários para fazer funcionar o popover. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

Por padrão o texto do popover é na cor preta, inserido em uma caixa na cor branca com uma barra cinza no topo para acomodar o título do popover, as bordas são arredondadas e posicionada à direita da palavra que dispara o popover.

É possível alterar a posição da caixa popover com uso do atributo `data-placement` e seus valores: `top`, `right`, `bottom` e `left`, sendo `right` o valor padrão. Mostramos a seguir as sintaxes para posicionamento.

```
<a href="#" data-toggle="popover" title="Fruta tropical">maçã</a>   // padrão à direita
<a href=”#” data-toggle=”popover” data-placement=“top" ...>maçã</a>   // no topo
<a href=”#” data-toggle=”popover” data-placement=“bottom" ...>maçã</a> // embaixo
<a href=”#” data-toggle=”popover” data-placement="left" ...>maçã</a>   // à esquerda
```

No exemplo a seguir mostraremos a criação de popover nas quatro posições possíveis.

- **HTML**

```
<h4>Popover à direita</h4>
<p>A <a href="#" data-toggle="popover" data-placement="right"
   title="Informações nutricionais"
data-content="O consumo ...">maçã</a> é uma das principais frutas do clima temperado.</p>
```

```
<h4>Popover à direita</h4>
<p>A <a href="#" data-toggle="popover" data-placement="top"
   title="Informações nutricionais"
data-content="O consumo ...">maçã</a> é uma das principais frutas do clima temperado.</p>

<h4>Popover à direita</h4>
<p>A <a href="#" data-toggle="popover" data-placement="bottom"
   title="Informações nutricionais"
data-content="O consumo ...">maçã</a> é uma das principais frutas do clima temperado.</p>

<h4>Popover à direita</h4>
<p>A <a href="#" data-toggle="popover" data-placement="left"
   title="Informações nutricionais"
data-content="O consumo ...">maçã</a> é uma das principais frutas do clima temperado.</p>
/* Mais quatro exemplos com popover em botões */
```

- **JavaScript**

```
<script type="text/javascript">
   $('[data-toggle="popover"]).popover();
</script>
</body>
</html>
```

[.../c5/popover-data-toggle.html]

Alerta: O funcionamento do componente popover requer a inclusão do plugins popover e *tooltip*.

O componente popover, mesmo quando criado com uso do atributo `data-toggle` precisa, obrigatoriamente, ser inicializado manualmente pelo desenvolvedor, conforme o código JavaScript mostrado, ou seja, deverá ser declarado o método `popover()` para o elemento que o dispara (no nosso exemplo `[data-toggle="popover"]`).

Ao visualizar este exemplo clique sobre a palavra maçã (cor azul por padrão) nos textos para ver o popover em ação nas suas quatro posições possíveis.

5.2.7.2 Popover com JavaScript nativo

Na verdade, nas opções com uso de `data-toggle` e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação

HTML e acionam os códigos JavaScript necessários para fazer funcionar o popover e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar o popover.

O método nativo para inicializar um popover tem a sintaxe conforme mostrada a seguir.

```
$('seletor').popover(options)
```

onde `seletor` é uma referência ao elemento container da palavra que dispara o popover.

As opções para esse método são:

Nome	Tipo	Default	Descrição
animation	boleano	true	Aplica o efeito fade ao mostrar/esconder o popover
html	boleano	false	O valor marcação HTML define que o conteúdo do popover será a marcação HTML aqui definida em lugar de texto simples.
placement	string ou função	'right'	Define o posicionamento do popover em relação à palavra que o dispara. Os valores possíveis são: top, right, bottom e left. Admite ainda a combinação de auto com um dos quatro posicionamentos, por exemplo: 'auto left' que posiciona à esquerda quando há espaço à esquerda e à direita caso contrário.
selector	string	false	Essa opção permite que se delegue a função de disparar o popover a um determinado seletor. Isso faz com que seletores inseridos no DOM dinamicamente herdem a funcionalidade popover.
title	string ou função	''	Com uso dessa opção podemos definir um título padrão para o popover. Para visualizar o título padrão é necessário que o atributo title não seja definido no elemento container do popover. Se for definido, sobrescreverá o título padrão.
trigger	string	'click'	O valor para essa opção define o evento que dispara o popover. Os valores possíveis são: click, hover, focus e manual. É admitido usar mais de um valor e nesse caso devem-se separar os valores com espaço. O valor manual não pode ser combinado com outro valor.
content	string ou função	''	Define o conteúdo descritivo do popover desde que o atributo data-content não tenha sido definido.
delay	número ou object	0	Define o tempo, em milissegundos, de retardo para abertura e fechamento do popover. Por exemplo, número: delay: 200 ou objeto: delay: { show: 100, hide: 200 }. Não se aplica para a opção trigger: "manual".

Nome	Tipo	Default	Descrição
container	string ou false	false	Define um container ao qual o popover será anexado. Por exemplo: container: '#janelaModal'.
template	string	'<div class="popover" role=" popover "> <div class=" popover-arrow"> </div> <div class=" popover-inner"> </div></div>'	Marcação HTML base a usar para criar um popover. O valor do atributo title será injetado no elemento com a classe popover-inner, a setinha será injetada no elemento com a classe popover-arrow. O container geral é no elemento com a classe .popover.
viewport	string ou object ou function	{ selector: 'body', padding: 0 }	Confina o popover aos limites do seu elemento.

São previstos ainda os métodos conforme as sintaxes mostradas a seguir.

```
$('seletor').popover( "show" )    // Mostra o popover
$('seletor').popover( "hide" )    // Esconde o popover
$('seletor').popover( "toggle" )  // Alterna mostra/esconde o popover
$('seletor').popover( "destroy")  // Esconde e destrói o popover
```

Os eventos nativos são descritos a seguir.

Evento	Descrição
show.bs.popover	Esse evento ocorre quando o usuário dispara o popover, mas o conteúdo dele ainda não foi mostrado.
shown.bs. popover	Esse evento ocorre quando o conteúdo do popover é mostrado.
hide.bs. popover	Esse evento ocorre imediatamente antes de o popover ser fechado.
hidden.bs. popover	Esse evento ocorre depois que o popover foi fechado.
inserted.bs. popover	Esse evento ocorre depois do evento show.bs.popover quando o template do popover for adicionado ao DOM.

A criação de popover com uso de JavaScript é idêntica à criação de tooltip com uso de JavaScript como mostrado no exemplo do item [5.2.6.2]. Baseado naquele exemplo, sugerimos ao leitor, como exercício, criar uma página exemplo para popover com JavaScript.

5.2.8 Alerta

Alerta é uma funcionalidade que permite ao desenvolvedor apresentar ao usuário uma caixa com mensagem de alerta.

A marcação típica para criar uma caixa de alerta consta de um `div` container ao qual se atribui a classe `alert`.

Dentro desse `div` marca-se um elemento `button` do tipo `button` com a classe `close`, o par atributo/valor igual a `data-dismiss="alert"` e a seguir a marcação do título e conteúdo da mensagem.

Essa marcação, conforme mostrado a seguir, cria uma caixa de alerta padrão ou de alerta geral, renderizada em tons de amarelo.

- **HTML**

```
<div class="alert">
   <button type="button" class="close" data-dismiss="alert">x</button>
   <h4>Alerta!</h4>
   <p>Você tem certeza que deseja deletar o arquivo?</p>
</div>
```

Para fazer funcionar o alerta o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap. Ao contrário dos componentes estudados anteriormente não há disparo de alerta com uso de atributos `data-*`.

O método nativo para inicializar um alerta tem a sintaxe conforme mostrada a seguir.

```
$('seletor').alert()
```

onde `seletor` é uma referência ao elemento container do alerta.

É previsto ainda o método conforme a sintaxe mostradas a seguir.

```
$('seletor').alert("close") // fecha a caixa de alerta
```

Os dois eventos nativos são descritos a seguir.

Evento	Descrição
close.bs.alert	Esse evento ocorre imediatamente após a chamada do método alert com a instância close.
closed.bs.alert	Esse evento ocorre ao final do fechamento da caixa de alerta (após terem sido completadas as transições CSS).

Para mais informações, uso e exemplos de caixas de alerta ver [4.14].

5.2.9 Botões

A estilização de botões com uso de classes (CSS) foi estudada no capítulo 3 nos itens [3.5.1.2], [3.5.1.3], [3.5.1.4] e o agrupamento de botões foi estudado no capítulo 4 no item [4.3]. Neste item do capítulo 5 estudaremos a manipulação de botões com uso de JavaScript.

São previstos três métodos para manipulação de botões conforme mostrados a seguir.

```
$(.botaoN).button('toggle') // alterna o estado do botão - ativo/repouso
$(.botaoN).button('reset')  // retorna o botão ao estado repouso
$(.botaoN).button('string') // altera o texto do botão
```

onde `.botaoN` identifica o botão ao qual será aplicado o método.

Alerta: O método $('`.botaoN).button('loading'`") está em desuso e será retirado na versão BS4.

No exemplo a seguir mostraremos a aplicação destes três métodos para manipulação de botões.

- **HTML**

```
<button type="button" id="botaoToggle"
   class="btn btn-default">Botão com button('toggle')</button>
<button type="button" class="btn btn-default">Botão sem button('toggle')</button></div>
<div id="botaoReset">
   <button type="button" class="btn btn-default">Padrão</button>
   <button type="button" class="btn btn-primary">Primário</button>
   <button type="button" class="btn btn-success">Sucesso</button>
   <button type="button" class="btn btn-info">Informação</button>
   <button type="button" class="btn btn-warning">Atenção</button>
   <button type="button" class="btn btn-danger">Perigo</button>
</div>
<button type="button" id="botaoString" class="btn btn-default"
   data-complete-text="Carregamento terminou!>Padrão</button>
```

- **JavaScript**

```
<script type="text/javascript">
// Ao clicar altera a aparência do botão para clicado
$("#botaoToggle").click(function(){
   $(this).button('toggle');
})
/* Ao clicar o texto do botão muda para "loading"
ao terminar o carregamento o texto volta ao valor inicial */
$("#botaoReset .btn").click(function(){
```

```
    $(this).button('loading').delay(1000).queue(function(){
        $(this).button('reset');
        $(this).dequeue();
    });
});
/* Ao clicar o texto do botão muda para "loading" ao terminar o carregamento o texto
muda para o valor do atributo data-complete-text (ver marcação HTML) */
$("#botaoString").click(function(){
    $(this).button('loading').delay(1000).queue(function() {
        $(this).button('complete');
        $(this).dequeue();
    });
});
</script>
</body>
</html>
```

[.../c5/botoes-javascript.html]

Ao visualizar este exemplo clique os botões para ver os métodos JavaScript em ação.

5.2.10 Collapse

Collapse é uma funcionalidade destinada a criar o efeito revelar/esconder para apresentação de conteúdos.

Existem duas maneiras de se projetar este efeito: diretamente com uso do par atributo/valor `data-toggle="collapse"` ou codificando com uso de métodos JavaScript previstos no Boostrap. Veremos a seguir essas duas maneiras.

5.2.10.1 Collapse com uso de data-toggle

Para criar este efeito usa-se uma marcação padrão com classes nativas do Bootstrap e a declaração do atributo `data-toggle="collapse"` para um elemento `a` ou `button` que deverá marcar o link que irá revelar/esconder os conteúdos.

Se for usado o elemento `a`, seu atributo `href` deverá ter valor igual ao valor do `id` do container dos conteúdos a revelar/esconder.

Se for usado o elemento `button` ele deverá receber o atributo `data-target` com valor igual ao valor do `id` do container dos conteúdos a revelar/esconder.

Os mecanismos internos do Bootstrap identificam os atributos `data-*` e as classes definidas na marcação e acionam os códigos JavaScript necessários para fazer funcionar o efeito. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

No exemplo a seguir mostraremos a criação do efeito revelar/esconder simples acionado por um link ou por um botão e o uso deste efeito para criar um menu acordeão com três links marcados com elementos `a` e ao qual atribuímos a `classe="btn"` fazendo com que os links fossem estilizados como um botão.

A marcação HTML para o exemplo é mostrada e comentada a seguir.

- **HTML**

```
// Exemplo de collapse simples
1. <a data-toggle="colapse"  href="#colapseUm">Maçã</a>
2. <button class="btn btn-primary" type="button" data-toggle="collapse"
      data-target="#colapseUm" aria-expanded="false">Maçã</button>
3. <div id="colapseUm" class="collapse">
4. <div class="well">
5.    <p><img src="collapse-maca.jpg" alt="maçã" class="pull-left">A maçã...</p>
6.    <small>Fonte: Wikipedia</small>
7. </div> <!-- /. well -->
8. </div> <!-- /#colapseUm -->
// Exemplo de collapse para criar menu acordeão
13. <div class="panel-group" id="accordion" role="tablist" aria-multiselectable="true">
14. <div class="panel panel-default">
15. <div class="panel-heading" role="tab" id="headingLaranja">
16.    <h4 class="panel-title">
17.       <a class="btn btn-info" role="button" data-toggle="collapse"
          data-parent="#accordion" href="#collapseLaranja" aria-expanded="true"
      aria-controls="collapseLaranja">Laranja</a>
18.    </h4>
19. </div>
20. <div id="collapseLaranja" class="panel-collapse collapse in" role="tabpanel"
      aria-labelledby="headingLaranja">
21.    <div class="panel-body">
22.       <p><img src="collapse-laranja.jpg" alt="laranja" class="pull-left">A laranja
   é ...</p><small>Fonte: Wikipedia</small>
23.    </div>
24. </div>
```

```
25. </div> <!-- /.panel -->
26. <div class="panel panel-default"> <!-- painel 2 --></div> <!-- /.panel -->
27. <div class="panel panel-default"> <!-- painel3 --></div> <!-- /.panel -->
28. </div> <!-- /.panel-group -->
```

[.../c5/collapse-data-toggle.html]

Alerta: O funcionamento do componente collapse requer a inclusão do plugins collapse e *transition*.

Código comentado:

Linha(s)	Descrição
Linhas 1 a 7	Marcação de um painel revelar/esconder simples.
Linha 1	Link para acionar o efeito revela/esconde. O par atributo/valor `data-toggle="collapse"` é obrigatório e o atributo `href` deve apontar para o `id` do container do conteúdo a revelar/esconder (linha 3).
Linha 2	Botão para acionar o efeito revelar/esconder. O par atributo/valor `data-target="colapseUm"` é obrigatório e deve ser igual ao valor do `id` do container do conteúdo a revelar/esconder (ver linha 3).
Linhas 3	Container geral para o conteúdo a revelar/esconder. É obrigatório definir, para esse container, a classe com valor `collapse`.
Linhas 4	Container que usa a classe `well` prevista no Bootstrap com a finalidade de criar e estilizar uma caixa com bordas arredondadas e cor de fundo (ver componente Well em [4.20]).
Linhas 5 e 6	Conteúdo a revelar/esconder.
Linhas 7 e 8	Tags de fechamento dos containeres well e geral.
Linhas 13 a 28	Marcação de três painéis revelar/esconder para criar o menu acordeão.
Linha 13	Container geral do menu que deverá, obrigatoriamente, receber a classe `panel-group` e um `id` de livre escolha. O valor do `id` será igual ao valor do atributo `data-parent` marcado em cada um dos links (ou botões) que acionam o efeito revelar/esconder. (ver linha 17)
Linhas 14 a 25	Container geral do primeiro painel que compõe o menu. Nas linhas 26 e 27 encontram-se os outros dois painéis cuja marcação HTML foi omitida por ser semelhante à deste primeiro painel.
Linha 15 a 19	Container geral do link ou botão destinado a revelar/esconder o conteúdo de cada painel. Notar que o valor do atributo `id` deste container deverá ser igual ao valor do atributo `aria-labelledby` do container geral do cojnteúdo a revelar/esconder (ver linha 20).
Linha 16 a 18	Elemento `h4` container do link para revelar/esconder.

Linha(s)	Descrição (cont.)
Linha 17	Container do link que aciona o revelar/esconder. O atributo `href` deste link aponta para o `id` do container geral do conteúdo a revelar/esconder (ver linha 20).
Linhas 20 a 24	Container geral do conteúdo a revelar/esconder. A classe `in` neste container destina-se a fazer com que este conteúdo seja apresentado à vista (revelado ou aberto) quando a página for carregada.
Linhas 21 a 23	Container corpo dos conteúdos a revelar/esconder. Notar que neste container pode-se inserir qualquer tipo da marcação HTML e no nosso exemplo mostramos uma imagem em cada painel, além de textos.

5.2.10.2 Collapse com JavaScript nativo

Na verdade, nas opções com uso de `data-*` e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para fazer funcionar o efeito revelar/esconder e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar o accordion.

O método nativo para inicializar o efeito tem a sintaxe conforme mostrada a seguir.

```
$('seletor').collapse()
```

onde `seletor` é uma referência ao elemento container do conteúdo a revelar/esconder.

As opções para esse método são:

Nome	Tipo	Default	Descrição
parent	selector	false	Define o elemento-pai para o efeito revelar/esconder.
toggle	boleano	true	Altera a visibilidade do conteúdo quando acionado.

São previstos ainda os métodos conforme as sintaxes mostradas a seguir.

```
$('seletor').collapse( "show")    // revela o conteúdo
$('seletor').collapse( "hide")    // esconde o conteúdo
$('seletor').collapse("toggle")   // alterna revelar/esconder o conteúdo
```

Os quatro eventos nativos são descritos a seguir.

Evento	Descrição
`show.bs.collapse`	Esse evento ocorre imediatamente após a chamada do método collapse com a instância show.
`shown.bs.collapse`	Esse evento ocorre ao final da abertura de um conteúdo (após terem sido completadas as transições CSS).
`hide.bs.collapse`	Esse evento ocorre imediatamente após a chamada do método hide().
`hedden.bs.collapse`	Esse evento ocorre ao final do fechamento de um conteúdo (após terem sido completadas as transições CSS).

A obtenção do efeito revelar/esconder com uso de JavaScript é idêntica à criação de tooltip com uso de JavaScript como mostrado no exemplo do item [5.2.6.2]. Baseado naquele exemplo, sugerimos ao leitor, como exercício, criar uma página exemplo para o efeiot revelar/esconder com JavaScript.

5.2.11 Carousel

Carousel é uma funcionalidade que se destina a criar um slideshow acionado com uso de um efeito semelhante a um carrossel.

Existem duas maneiras de se projetar o efeito carrossel: diretamente com uso de marcação padrão e de atributos classe e `data-ride` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

Todas as imagens a serem mostradas no slideshow carrossel devem ter as mesmas dimensões (largura x altura). A largura deverá ser igual ou maior a maior largura assumida pelo container do carrossel e a altura é de livre escolha do autor, isto é, o Bootstrap não estabelece uma razão de aspecto para as imagens.

Por exemplo: se o carrossel foi projetado para ocupar toda a largura de uma linha (`div.row`) as imagens deverão ter uma largura total igual (ou maior) do que 1170px pois esta é a maior largura de uma linha para resoluções acima de 1200px que é o maior breakpoint do Bootstrap. Se o carrossel foi projetado para ocupar a metade de uma linha a largura das imagens deverá ser igual à largura correspondente à união de seis colunas adjacentes (aproximadamente 600px), pois uma linha contém 12 colunas.

5.2.11.1 Carousel com uso de data-ride

Para criar o efeito carrossel em um slideshow usa-se uma marcação padrão com classes nativas do Bootstrap e a declaração do atributoo `data-ride` e de classes.

Os mecanismos internos do Bootstrap identificam os atributos e as classes definidas na marcação e acionam os códigos JavaScript necessários para fazer funcionar o carrossel. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

No exemplo a seguir mostraremos a criação de um slideshow com uso do plugin Carousel.

- **HTML**

```
1.  <div id="meuCarrossel" class="carousel slide" data-ride="carousel">
2.  <ol class="carousel-indicators"> <!-- "Bolinhas" da navegação pelos slides -->
3.      <li data-target="#meuCarrossel" data-slide-to="0" class="active"></li>
4.      <li data-target="#meuCarrossel" data-slide-to="1"></li>
5.      <!-- mais três indicadores -->
6.      <li data-target="#meuCarrossel" data-slide-to="5"></li>
7.  </ol>
8.  <div class="carousel-inner" role="listbox"> <!-- Container dos slides -->
9.      <div class="item active">
10.     <img src="c1.jpg" alt="">
11.         <div class="carousel-caption">
12.             <h4>Slide 1</h4>
13.             <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.</p>
14.         </div>
14.     </div>
15.     <div class="item">
16.     <img src="c2.jpg" alt="">
17.         <div class="carousel-caption">
18.             <h4>Legenda 2</h4>
19.             <p>Curabitur hendrerit erat ut augue. Cras gravida.</p>
20.         </div>
21.     </div>
22.     <div class="item">
23.         <!-- seguem mais quatro slides -->
22.     </div>
23. </div> <!-- /.carousel-inner -->
```

```
24.     <a class="left carousel-control" href="#meuCarrossel" data-slide="prev" role="button">
        <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
           <span class="sr-only">Anterior</span></a>
25.     <a class="right carousel-control" href="#meuCarrossel" data-slide="next"
           role="button" >
     <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
     <span class="sr-only">Próximo</span>
     </a>
26. </div> <!-- /#meuCarrossel -->
```

[.../c5/carrossel-data-ride.html]

Código comentado:

Linha(s)	Descrição
Linhas 1 e 26	Container geral do carrossel. Deverá receber a classe de estilização prevista no Bootstrap denominada `carousel`. Adicionalmente declare a classe `slide` para permitir a transição dos slides com o efeito de movimentar lateralmente. Se essa classe não for definida a transição será abrupta. Definir também o atributo `id` com valor de livre escolha do autor. Esse valor servirá de referência para definir o valor do atributo `data-target` conforme mostrado nas linhas 3 a 6 e também será o valor do atributo target dos links "Anterior" e "Próximo" (ver linhas 24 e 25). No nosso exemplo escolhemos o valor `id="meuCarrossel"` para o container geral do carrossel (ver linha 1).
Linhas 2 a 7	Lista destinada a criar os indicadores de imagem ativa (as bolinhas de navegação pelas imagens). O quantidade de itens `li` nesta lista é igual à quantidade de slides. Na marcação desta lista definir, tal como mostrado na marcação, os atributos `class`, `data-target` e `data-slide-to`. O valor do atributo `data-slide-to` deverá ser uma sequência númerica iniciando em zero.
Linhas 8 e 23	Container interno para o carrossel. Definir a classe `carousel-inner`.
Linhas 9 e 14	Container para cada imagem e sua legenda. Definir a classe `item` e a classe `active` para determinar a primeira imagem a aparecer no carrossel (imagem de capa).
Linhas 11 e 14	Container geral para a legenda da imagem. Definir, para esse container, a classe `carousel-caption`.
Linhas 24 e 25	Cria as setas para manualmente acionar o carrossel para a esquerda ou direita.

Na marcação HTML que demonstra este exemplo uma folha de estilos incorporada com apenas duas regras de estilos como mostrada a seguir foi comentada. Estas regras de estilos criam um efeito na legenda dos slides. Para visualizar o efeito, descomente as regras.

- **CSS**

```
<style rel="stylesheet">
h2 {font-size:20px;}
.carousel-caption {
   bottom: 0;
   right: 0;
   left: 0;
   padding: 0 5px 30px 5px;
   background: rgba(0,0,0,.6);
}
.carousel-indicators { bottom: 0; }
</style>
```

5.2.11.2 Carousel com JavaScript nativo

Na verdade, nas opções com uso de marcação padrão e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para fazer funcionar o carrossel e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para acionar o carrossel.

O método nativo para inicializar um carrossel tem a sintaxe conforme mostrada a seguir.

```
$('seletor').carousel()
```

onde `seletor` é uma referência ao elemento container do carrossel.

As opções para esse método são:

Nome	Tipo	Default	Descrição
interval	número	5000	Tempo de espera em milissegundos para haver a transição automática para a próxima imagem. Definir o valor false para essa opção interrompe a transição automática.
pause	string	'hover'	Ação do usuário necessária para pausar o slideshow. A ação padrão ('hover') é colocar o mouse sobre a imagem para pausar e retirar o mouse para continuar.
wrap	boleano	'true'	Define se a movimentação dos slides será infinita ou apenas um ciclo e parar.
keybord	boleano	'true'	Define se os comandos avançar/recuar dos slides serão ou não controlados pelas setas do teclado.

São previstos ainda os métodos conforme as sintaxes mostradas a seguir.

```
$('seletor').carousel( "cycle")// movimenta os slides da esquerda para a direita
$('seletor').carousel( "pause")// pausa o movimento dos slides
$('seletor').carousel(number)  // movimenta os slides para o slide number (0,1,2,3 etc.)
$('seletor').carousel("prev")  // movimenta para o slide anterior ao atual
$('seletor').carousel("next")  // movimenta para o slide posterior ao atual
```

Os dois eventos nativos são descritos a seguir.

Evento	Descrição
slide.bs.carousel	Esse evento ocorre no início da movimentação dos slides.
slid.bs.carousel	Esse evento ocorre ao final da transição do slide.

Nestes dois eventos estão disponíveis as propriedades `direction` e `event.relatedTarget` que retornam a direção de movimentação (left ou right) e o elemento, respectivamente, do slide que se tornou ativo.

No exemplo a seguir usamos a mesma marcação do exemplo mostrado no item anterior [5.2.11.1], retirando apenas o atributo `data-ride` da marcação HTML e criamos comandos para o slideshow usando algumas das funcionalidades JavaScript aqui estudadas. O script constante do exemplo é mostrado a seguir.

- **JavaScript**

```
<script type="text/javascript">
$('#meuCarrossel').carousel(
   {
      interval: 500,
      wrap: false,
      keyboard: false
   });
$('#meuCarrossel').on('slide.bs.carousel', function () {
   console.log('Início da movimentação do slide');
})
$('#meuCarrossel').on('slid.bs.carousel', function () {
   console.log('Fim da movimentação do slide');
});
</script>
```

[.../c5/carrossel-javascript.html]

Sugerimos ao leitor, como exercício, criar uma cópia do arquivo deste exemplo e explorar as demais funcionalidades JavaScript para o plugin carousel escrevendo seus próprios scripts.

5.2.12 Affix

Affix é uma funcionalidade destinada a manter um container fixo em uma posição na página, ou seja, se o usuário aciona a barra de rolagem da página o container não se movimenta.

Existem duas maneiras de se projetar o efeito affix: diretamente com uso de marcação padrão e de atributos `data-spy` ou codificando com uso de métodos JavaScript previstos no Bootstrap. Veremos a seguir essas duas maneiras.

5.2.12.1 Affix com uso de data-spy

Para criar o efeito affix usa-se uma marcação padrão com declaração do atributos `data-spy`.

Os mecanismos internos do Bootstrap identificam os atributos definidos na marcação e acionam os códigos JavaScript necessários para manter o container fixo na página. O desenvolvedor não precisa criar código JavaScript, eles são criados automaticamente pelo Bootstrap.

No exemplo a seguir mostraremos a criação de um menu fixo na tela com uso da funcionalidade affix.

- **HTML**

```
<ul class="nav nav-tabs nav-stacked" data-spy="affix" data-offset-top="265">
   <li><a href="#tp">Topo</a></li>
   <li><a href="#um">Link 1</a></li>
   <li><a href="#dois">Link 2</a></li>
   <li><a href="#tres">Link 3</a></li>
   <li><a href="#quatro">Link 4</a></li>
   <li><a href="#cinco">Link 5</a></li>
</ul>
<h2 id="um">Link 1</h2>
   <p>Lorem ipsum...</p>
<h2 id="dois">Link 2</h2>
   <p>Pellentesque sapien...</p>
```

```
<h2 id="tres">Link 3</h2>
   <p>Nulla facilisi...</p>
<h2 id="quatro">Link 4</h2>
   <p>Curabitur hendrerit...</p>
<h2 id="cinco">Link 5</h2>
   <p>Cum sociis natoque...</p>
```

[.../c5/affix-data-spy.html]

Definimos o atributo `data-spy="affix"` para o container do elemento a manter fixo na página (no nosso exemplo o menu) e o atributo `data-offset-top` que define a quantidade de pixels a rolar a partir da qual o container será mantido fixo na sua posição original.

Pode-se usar qualquer tipo de dispositivo de navegação, previsto ou não no Bootstrap, contudo essa funcionalidade não tem uma estilização padrão nativa. O autor deverá criar uma folha de estilos conforme suas necessidades. Neste exemplo a folha de estilos está incorporada à marcação da página.

Consulte o exemplo, verifique o funcionamento e depois comente a folha de estilos voltando a verificar o funcionamento.

5.2.12.2 Affix com JavaScript nativo

Na verdade, nas opções com uso de marcação padrão e com uso de JavaScript o mecanismo do plugin usa JavaScript. A diferença é que para a primeira opção são os mecanismos internos do Bootstrap que identificam os atributos da marcação HTML e acionam os códigos JavaScript necessários para tornar o container fixo e na opção com JavaScript nativo o desenvolvedor precisa definir manualmente os métodos JavaScript próprios do Bootstrap para fixar o container.

O método nativo para inicializar um affix tem a sintaxe conforme mostrada a seguir.

```
$('seletor').affix()
```

onde `seletor` é uma referência ao elemento container a manter fixo.

A duas opções para esse método é:

Nome	Tipo	Default	Descrição
offset	número ou função ou objeto	10	Define a posição, em número de pixels, do container (no nosso exemplo um menu) fixo em relação ao topo da tela. As coordenadas são top, right, bottom e left. Por exemplo: offset: { top:20 }.

Nome	Tipo	Default	Descrição
target	seletor ou nó ou elemento jQuery	O objeto window	Define o elemento em relação ao qual o container (no nosso exemplo um menu) será fixo.

No exemplo a seguir usamos a mesma marcação do exemplo mostrado no item anterior [5.2.12.1], retirando apenas os atributos `data-spy` e `data-offset` da marcação HTML e criamos comandos para o acionar o affix usando algumas das funcionalidades JavaScript aqui estudadas. A retirada dos atributos da marcação HTML e o script constante do exemplo é mostrado a seguir.

- **HTML**

```
<ul class="nav nav-tabs nav-stacked" data-spy="affix" data-offset-top="265">
```

Nessa opção não se usam os atributos `data-*`.

- **JavaScript**

```
<script type="text/javascript">
   var offset = $('.nav').position().top;
   $('.nav').affix({
      offset: offset
   });
</script>
```

[.../c5/affix-javascript.html]

Notar que com uso de JavaScript podemos inspecionar dinamicamente as coordenadas do container (no nosso exemplo um menu), com uso do método `position()` da jQuery, e defini-la com uso da opção `offset`.

CAPÍTULO 6

Criando um layout com Bootstrap

Neste capítulo, iremos aplicar as funcionalidades do framework para criar um layout a partir da imagem fornecida pelo designer do site.

6.1 Introdução

É quase certo que as regras-padrão de estilização previstas na folha de estilo do Bootstrap não sejam adequadas ao design do site que você pretende desenvolver. Existem três opções básicas de personalizar a folha de estilo do Bootstrap para atender às necessidades específicas de um projeto.

A primeira opção é fazer o download personalizado no site do Bootstrap em *http://getbootstrap.com*, a segunda é alterando as variáveis LESS de acordo com suas necessidades e a terceira opção é fazer o download completo do framework Bootstrap (foi a opção que adotamos para criar os exemplos deste livro) e projetar uma folha de estilo personalizada para as necessidades do projeto, conforme mencionamos no item 1.3, quando dissemos que era sugerido que essa folha de estilo fosse nomeada como *custom.css*. Tal folha deverá ser vinculada ao documento depois da folha de estilos do Bootstrap, conforme mostrado a seguir.

```
...
   <title>Título da página</title>
   <link rel="stylesheet" href="css/bootatrap.min.css">
   <link rel="stylesheet" href="css/custom.css">
...
```

A finalidade desta folha de estilo é a de sobrescrever os estilos-padrão do Bootstrap, adaptando-os às necessidades do projeto. Veja o exemplo a seguir.

Vimos no item 3.1 que a marcação HTML para criar um botão é a mostrada a seguir.

```
<button type="button" class="btn btn-default">Botão básico</button>
```

Esta marcação cria um botão básico que é renderizado com cor de fundo branca, borda cinza arredondada e texto na cor preta. No estado hover a cor de fundo do botão é cinza. Abrindo a folha de estilos *bootstrap.css* localizada na pasta *css* do Bootstrap em editor de texto e nela localizando os seletores `.btn` e `.btn-default` encontramos as declarações CSS que criam (estilizam) o botão básico. No trecho de código a seguir, retirado daquela folha de estilo, mostramos as regras CSS para aqueles seletores.

```
.btn {
  display: inline-block;
  padding: 6px 12px;
  font-size: 14px;
  background-image: none;
  border: 1px solid transparent;
  border-radius: 4px;
  /* mais declarações CSS */
}
.btn-default {
  color: #333;
  background-color: #fff;
  border-color: #ccc;
}
```

Conforme estudado neste livro, sabemos que existem várias classe nativas destinadas a estilizar os botões em diferentes cores e dimensões.

Suponha que no seu projeto os botões básicos foram definidos pelo designer com a cor de fundo preta (black), a cor do texto verde (lime) e borda na cor vermelha (red) com cantos vivos.

Neste caso, para personalizar os botões, basta que você declare, no arquivo *custom.css*, regras CSS para atender aquela estilização, conforme mostrado a seguir.

```
.btn-default {
  color: lime;
  background-color: black;
  border-color:red;
  border-radius:0;
}
```

Assim, você criou um conflito de regras CSS contidas em dois arquivos separados servidos à mesma página. Resolve-se conflito de regras CSS aplicando-se os critérios da especificidade e do efeito cascata respectivamente, ou seja, vence a regra mais específica, e se houver empate, como é o caso do nosso exemplo, vence o efeito cascata.

Como lincamos à página o arquivo *app.css* depois do arquivo *bootstrap.min.css*, a regra contida naquele arquivo é a vencedora e o botão básico terá uma estilização personalizada.

Consulte o arquivo *[.../c6/botao-basico-personalizado.html]* para verificar essa personalização do botão básico.

Neste capítulo, iremos criar um layout personalizando com uso de uma folha de estilos do tipo *custom.css*; contudo, para facilitar o estudo, a folha de estilo será incorporada na seção `head` da página e não lincada, como deve ser em um caso real.

Uma ferramenta indispensável para criação de uma folha de estilos personalizada é o console do navegador, que poderá ser instalado como complemento, como é o caso do Firebug, ou ser nativo do navegador, como é o caso da maioria dos navegadores modernos. Não é do escopo deste livro ensinar o uso de consoles de navegador, mas o leitor necessariamente deverá saber usar pelo menos os painéis de inspeção e edição do HTML e das CSS integrantes do console.

Dica: O Zeno Rocha criou vídeos no YouTube que ensinam de forma bem didática como usar o console do navegador. Assista os vídeos em *http://kwz.me/JR*.

O console do navegador mostra ao autor quais são as regras CSS constantes da folha de estilo do Bootstrap aplicadas em determinado elemento da marcação HTML e consequentemente qual seletor usar na personalização.

6.2 Layout iMasters

Para ilustrar este capítulo, usaremos parte do layout da Home Page do iMasters (*http://kwz.me/vZ*) com o aspecto que ela apresentava há um tempo atrás e que pode ser visualizada na figura 6.1.

Note na figura que a largura máxima de alguns conteúdos da página é fixa (940px) e que a cor de fundo de algumas barras do layout ocupam 100% da janela do navegador.
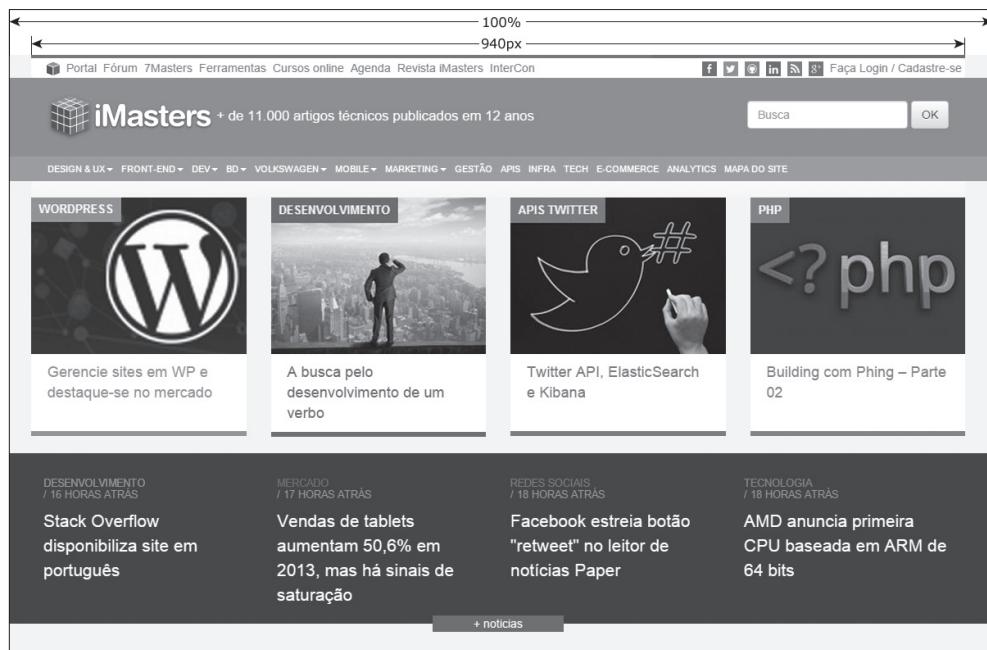
Figura 6.1 – Home page iMasters (parte).

Consideramos para o propósito deste capítulo interromper o layout no final da barra superior das notícias como mostrado na figura. Trata-se apenas do início do layout, mas apresenta uma configuração compatível e suficiente para cumprir os objetivos deste capítulo. Fica como sugestão ao leitor, a título de exercício e fixação dos seus estudos, criar as partes do layout abaixo da barra de notícias.

Todas as imagens do layout estão disponíveis na pasta *imagens* que se encontra dentro da pasta *imasters* no diretório *c6* na pasta do download dos arquivos de exemplos deste livro.

Antes de prosseguir, consulte o arquivo *[.../c6/imasters/imasters.html]* e visualize a home page completa do site do iMasters, da qual foi extraída a imagem mostrada na figura 6.1 e que servirá de base para as técnicas estudadas neste capítulo.

Conforme estudamos no item [3.1], o Bootstrap prevê um sistema de grid para construção de layout que por padrão se estende até uma largura máxima igual a 1170px em larguras de viewport maiores do que 1200px. O espaçamento entre cada coluna é obtido com a definição de um `padding` lateral (à esquerda e à direita) igual a 15px, resultando em um espaçamento entre colunas igual a 30px. Assim, somados o `padding` lateral esquerdo da primeira coluna e o `padding` lateral direito da última coluna, a largura máxima para o conteúdo do layout é de 1170px – 30px = 1140px.

A figura 6.2 apresenta um esquema mostrando as dimensões horizontais do grid--padrão, conforme descrito.
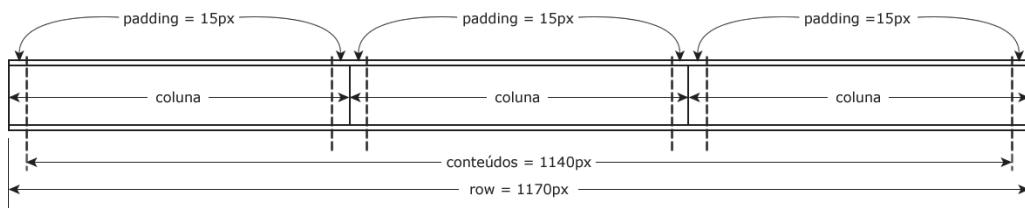
Figura 6.2 – Esquema do grid-padrão.

É fácil personalizar a largura de um layout para diferentes breakpoints, pois, conforme estudamos no item 3.1.1 uma aplicação desenvolvida com Bootstrap deverá estar contida em um elemento container geral que recebe a classe de nome `container`. É esse container que define a largura máxima do layout para os diferentes breakpoints.

Inspecionando a folha de estilos do Bootstrap, encontramos as seguintes regras CSS para o seletor de classe `.container`.

```
@media (min-width: 768px) {
   .container {
   width: 750px;
   }
}
@media (min-width: 992px) {
   .container {
   width: 970px;
   }
}
@media (min-width: 1200px) {
   .container {
   width: 1170px;
   }
}
```

Então, para definir uma largura de conteúdo máxima para o layout igual a 940px (largura máxima do layout do iMasters), basta criar a seguinte regras CSS na folha de estilos personalizada da nossa aplicação.

```
@media (min-width: 1200px) {
   .container {
   width: 970px; /* 970px - 30px = 940px */
   }
}
```

Alerta: Em nosso exemplo, não iremos alterar a largura máxima do layout para 940px como é a largura do site do iMasters, pois não existe qualquer prejuízo em se adotar a largura máxima padrão do Bootstrap que é de 1130px.

Por razões didáticas, vamos mostrar a criação do layout por etapas, conforme descritas nos itens que se seguem.

6.2.1 Primeira etapa: navegação superior

O template para o desenvolvimento do layout é conforme estudamos no item [1.4], contudo por tratar-se de personalização não usaremos o arquivo *bootstrap-theme.css*.

No topo do layout existe um mecanismo de navegação constante de links para seções do site, links para redes sociais em forma de ícones e link para login e cadastro. Na parte superior deste mecanismo de navegação existe uma barrinha superior decorativa com três seções em cores diferentes. Note que a cor de fundo da página é cinza-clara.

A barrinha superior decorativa será construída com uso de um elemento `div` com a classe `barrinha-superior` ao qual será aplicada uma cor de fundo constituída por um gradiente CSS, conforme os códigos mostrados a seguir.

- **HTML**

```
<body>
<div class="container">
   <div class="barrinha-superior"></div>
...
```

Notar que não há necessidade de se criar uma linha (`<div class="row">...</div>`) para inserção da barrinha superior, ela pode ser elemento-filho do container geral `div.container`.

- **CSS**

```
body { background: #f5f5f5; } /* cor de fundo da página */
.barrinha-superior {
   height: 4px;
   background: #369caf; /* fallback para navegadores que não suportam gradientes */
   background-image: linear-gradient( to right, #369caf 33.33%, #e37900 33.33%,
                     #e37900 66.66%, #af2d22 66.66%, #af2d22 );
   /* use a mesma declaração com uso de prefixos proprietários se for o caso */
}
```

Alerta: Notar que embora a barrinha superior seja constituída por três faixas sequenciais de cores sólidas diferentes as funcionalidades dos gradientes CSS possibilitam a obter aquele efeito.

Para criar o mecanismo de navegação superior, usaremos o componente do Bootstrap denominado barra de navegação básica estudada no item [4.7.1], e marcada conforme o código mostrado a seguir.

- **HTML**

```
<div class=”row”>
<div class="col-md-12">
<nav class="navbar navbar-default" id="nav-superior">
          <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
                 data-target="#barraSuperior">
              <span class="sr-only">Toggle navigation</span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
              <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand hidden-lg hidden-md hidden-sm" href="#"
              data-toggle="collapse" data-target="#barraSuperior">Menu</a>
          </div>
          <div class="collapse navbar-collapse" id="barraSuperior">
            <ul class="nav navbar-nav">
      <li><a href="#"><img src="imagens/icon-navsup.png" alt="logotipo do site"></a></li>
              <li><a href="#">Portal</a></li>
              <!-- mais sete itens -->
            </ul>
```

```
            <ul class="nav navbar-nav navbar-right">
              <li><a href="#" title="facebook"><img src="imagens/facebook.png"
                 alt="facebook"></a></li>
              <!-- mais cinco itens -->
              <li><a href="#">Faça Login / Cadastre-se</a></li>
            </ul>
          </div><!-- /.navbar-collapse -->
      </nav>
</div>
</div> <!-- /.row -->
```

Alerta: Notar que para o container geral da barra de navegação definimos o de nossa escolha (não padrão do Bootstrap) `id="nav-superior"`. Fizemos isso para diferenciar da outra barra de navegação que existe no layout, como veremos adiante.

[.../c6/imasters1.html]

Nesta etapa, a barrinha de topo estará estilizada e a barra de navegação será renderizada com a estilização padrão do Bootstrap.

Observe na figura 6.3 o resultado da primeira etapa.
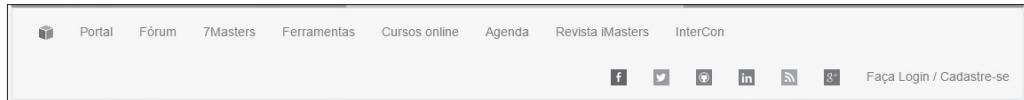
Figura 6.3 – Primeira etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout.

6.2.2 Segunda etapa: personalização da barra de topo

Ao término da primeira etapa, construímos a barrinha colorida superior e a barra de navegação estilizada segundo o padrão de estilos do Bootstrap.

Nesta etapa, criaremos as regras de estilo personalizadas, conforme mostradas e comentadas a seguir.

- **CSS**

```
1. #nav-superior { background:#fff; }
```

```
2. #nav-superior li a { padding:2px 4px; }
3. #nav-superior li a:hover { color:#00f; }
4. @media (min-width: 1px) {
5.     .navbar {
6.         min-height:auto;
7.         margin:0;
8.         border:none;
9.     }
10.    .navbar li a { color:#333; }
11. }
```

[.../c6/imasters2.html]

Código comentado:

Linha	Descrição
Linha 1	Esta declaração sobrescreve a cor de fundo padrão (#f8f8f8) da barra de navegação e a faz branca.
Linha 2	Sobrescrevemos o padding padrão do Bootstrap para os elementos `a` da barra de navegação que é maior do que aquele previsto para o nosso layout.
Linha 3	Definimos a cor azul, prevista para o estado over dos links.
Linha 4	Declarar `@media (min-width:1px)` tem o efeito de sobrescrever as regras de estilo padrão para todas as larguras de viewport. Se não tivéssemos lançado mão deste artifício as regras CSS contidas nas linhas 5 a 11 valeriam apenas para larguras de viewport menores que 768px (lembra-se do princípio mobile-first do desenvolvimento das CSS do Bootstrap?).
Linhas 5 a 10	Examinando a folha de estilo do Bootstrap encontramos as regras CSS para estilizar as barras de navegação. Ali estão as declarações para bordas cinzas e arredondadas, altura mínima de 50px, margens conforme mostrado na figura 6.3. As declarações CSS contidas nestas linhas estilizam a barra de navegação superior de acordo com o layout do iMasters.
Linha 11	Estiliza os links da barra de navegação na cor preta.

Observe na figura 6.4 o resultado da segunda etapa.

Figura 6.4 – Segunda etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout.

6.2.3 Terceira etapa: barra do logotipo e caixa de busca

Nesta etapa, iremos criar e personalizar a barra azul que contém o logotipo e a caixa de busca do site. Para esta barra definimos um atributo `id` com valor `barra1`.

A marcação HTML para esta barra é conforme mostrada a seguir.

- **HTML**

```
1.  <div class="row">
2.  <div class="col-lg-12">
3.  <div id="barra1">
4.      <img src="imagens/branding.png" alt="logo iMasters">
5.      <span class="texto-logo">+ de 11.000 artigos técnicos publicados em 12 anos</span>
6.      <form class="navbar-form navbar-right" role="search">
7.          <div class="form-group">
8.              <input type="text" class="form-control" placeholder="Busca">
9.          </div>
10.     <button type="submit" class="btn btn-default">OK</button>
11.     </form>
12.  </div> <!-- /#barra1 -->
13. </div>
14. </div> <!-- /.row -->
```

Código comentado:

Linha	Descrição
Linhas 1 e 14	Deninimos uma linha para container da barra.
Linhas 2 e 13	Criamos uma coluna dentro da linha container da barra.
Linhas 3 e 12	Definimos um container para o logotipo e o formulário com a finalidade de termos uma referência única para estilização.
Linha 4	Insere a imagem do logotipo do iMasters.
Linha 5	Marcação do texto ao lado do logotipo. Inserimos o texto dentro do container `span.texto-logo`, pois iremos estilizá-lo de forma diferenciada, como veremos adiante.
Linhas 6 a 11	Marcamos o formulário de busca conforme estudado no item [4.7.3].

- **CSS**

```
body { background: #f5f5f5 url(imagens/bg-header.png) repeat-x; }
#barra1 { background: #64a7b7; padding: 20px; }
.texto-logo { font-size: 120%; color:#fff; }
```

[.../c6/imasters3.html]

Essas três regras CSS são suficientes para estilizar a barra do logotipo e caixa de busca. Notar que para estender a cor de fundo da barra usamos uma imagem, com repetição no eixo dos x, como fundo do elemento `body`.

Observe na figura 6.5 o resultado da terceira etapa.
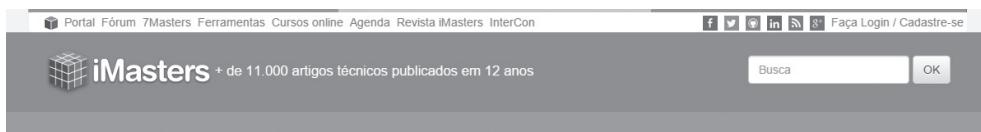
Figura 6.5 – Terceira etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout.

6.2.4 Quarta etapa: mecanismo de navegação principal

Nesta etapa, iremos criar e personalizar o mecanismo de navegação principal do site.

Para criar este mecanismo de navegação, usaremos o componente do Bootstrap denominado barra de navegação com link dropdown estudada no item [4.7.5], e marcada conforme o código mostrado a seguir.

- **HTML**

```
<div class="row">
<div class="col-lg-12">
   <nav class="navbar navbar-default">
   <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
         data-target="#barraPrincipal">
      <span class="sr-only">Toggle navigation</span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
         <span class="icon-bar"></span>
      </button>
   <a class="navbar-brand hidden-lg hidden-md hidden-sm"
      href="#" data-toggle="collapse" data-target="#barraPrincipal">Menu</a>
   </div>
   <div class="collapse navbar-collapse" id="barraPrincipal">
```

```
        <ul class="nav navbar-nav">
           <li class="dropdown">
     <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
                 aria-expanded="false">Design &#038; UX<span class="caret"></span></a>
              <ul class="dropdown-menu" role="menu">
                 <li><a href="#">User Experience</a></li>
                 <li><a href="#">Arquitetura de Informação</a></li>
                 <!-- mais itens -->
              </ul>
           </li>
           <!-- mais dropdown -->
           ...
           <li><a href="#">Mapa do site</a></li>
        </ul>
     </div><!-- /.navbar-collapse -->
     </nav>
  </div>
  </div> <!-- /.row -->
```

A marcação HTML mostrada é típica para criação de uma barra de navegação com links em menu dropdown como estudado no item [4.7.5].

- **CSS**

```
1. #barraPrincipal { margin-bottom:20px; background:#7bb3c1!important; }
2. #barraPrincipal li a {
3.     font-size:80%;
4.     text-transform:uppercase;
5.     padding:5px;
6.     color:#fff;
7. }
8. a:hover { text-decoration:none; }
9. #barraPrincipal li ul li a { color:#333; }
```

[.../c6/imasters4.html]

Código comentado:

Linha	Descrição
Linha 1	Definimos uma margem abaixo da barra de navegação e a cor de fundo azul-clara para a barra de navegação.
Linha 2 a 7	Estilizamos os links da barra de navegação.
Linha 8	Define a cor de fundo azul-clara para a barra.

Linha	Descrição (cont.)
Linha 9	Para os links do submenu, cujo fundo é branco, definimos uma cor escura para sobrescrever a cor branca definida para os links de abertura do submenu que estão em fundo azul.

Observe na figura 6.6 o resultado da quarta etapa.
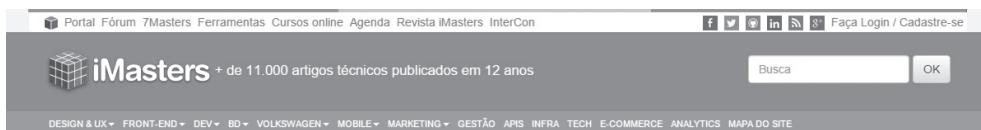
Figura 6.6 – Quarta etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout. Há inconsistências que serão tratadas nas etapas seguintes.

6.2.5 Quinta etapa: destaques da semana

Nesta etapa, iremos criar e personalizar os quatro blocos que promovem os artigos em destaque, publicados na semana, em diversas áreas do site.

Para alcançar os objetivos desta etapa, iremos usar uma linha do layout contendo quatro colunas em larguras de viewport maiores e dispostas em duas ou uma coluna em viewports menores.

A marcação HTML é mostrada a seguir.

- **HTML**

```
<section class=”row destaques”>
   <div class="col-sm-3 col-xs-6">
   <figure><a href="#" title="Gerencie sites em WP e destaque-se no mercado">
      <img src="imagens/banner-para-anuncios.jpg" class="img-responsive"
         alt="banner para anúncios"></a>
      <figcaption class="bc1"><a href="#"
         title="Gerencie sites em WP e destaque-se no mercado" class="cc1">
         Gerencie sites em WP e destaque-se no mercado</a>
      </figcaption>
   </figure>
   </div>
   <div class="col-sm-3 col-xs-6">
```

```
        <!-- semelhante ao anterior -->
    </div>
    <div class="col-sm-3 col-xs-6">
        <!-- semelhante ao anterior -->
    </div>
    <div class="col-sm-3 col-xs-6">
        <!-- semelhante ao anterior -->
    </div>
    </div>
</section> <!-- /.row -->
```

No código anterior, mostramos a marcação HTML completa somente para o primeiro bloco, abreviando-a para os três seguintes, pois a marcação para eles é semelhante à do primeiro item, exceto as classes `bc1` e `cc1` que para estes três blocos são: `bc2` e `cc2`, `bc3` e `cc3` e `bc4` e `cc4` respectivamente. Estas classes se destinam a estilizar cor de fundo e cor de textos para cada um dos blocos.

Notar ainda a definição de colunas em `<div class="col-sm-3 col-xs-6">` destinada a posicionar os quatro blocos em uma linha, duas linhas e quatro linhas conforme a largura da viewport.

- **CSS**

```
1.  .destaques img { width:100%; height: auto; }
2.  .destaques figcaption {
3.          font-size:130%;
4.          background:#fff;
5.          min-height:100px;
6.          padding:10px 20px;
7.  }
4.  .bc1 { border-bottom: 6px solid #e1734a; }
5.  .cc1 { display:block; color: #e1734a; }
6.  .bc2 { border-bottom: 6px solid #8d9b31; }
7.  .cc2 { display:block; color: #8d9b31; }
8.  .bc3 { border-bottom: 6px solid #4476ab; }
9.  .cc3 { display:block; color: #4476ab; }
10. .bc4 { border-bottom: 6px solid #9c6496; }
11. .cc4 { display:block; color: #9c6496; }
```

[.../c6/imasters5.html]

Código comentado:

Linha	Descrição
Linha 1	Por padrão as imagens inseridas na marcação HTML não são responsivas. Assim, declaramos a largura em 100% com a finalidade de fazê-las responsivas.
Linha 2 a 7	Regras para estilizar o container dos blocos.
Linhas 4 a 11	Classes para estilizar as cores de fundo e dos textos em cada um dos quatro blocos.

Observe na figura 6.7 o resultado da quinta etapa.

Figura 6.7 – Quinta etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout.

6.2.6 Sexta etapa: barra de notícias

Nesta etapa, iremos criar e personalizar a barra de notícias com fundo na cor preta que se estende por toda a largura da viewport. Essa barra contém quatro notícias com texto na cor branca, sendo que para cada uma das notícias há um título em cor diferente. Sobre a margem inferior da barra há um link em uma pequena caixa com o texto +notícias.

Para alcançar os objetivos desta etapa, a marcação HTML escolhida foi aquela que cria uma linha com quatro colunas, cada uma contendo uma notícia. O link +notícias esta contido dentro da linha.

A marcação HTML para esta etapa é mostrada a seguir.

- **HTML**

```
<div class=”container-fluid”>
   <section class="row noticias">
      <div class="container">
      <div class="col-sm-3">
         <h3><span class="cc1">Desenvolvimento</span> / 16 horas atrás</h3>
         <p><a href="#">Stack Overflow disponibiliza site em português</a></p>
      </div>
      <div class="col-sm-3">
         <h3><span class="cc2">Mercado</span> / 17 horas atrás</h3>
         <p><a href="#">Vendas de tablets aumentam 50,6% em 2013, mas há sinais de
saturação</a></p>
      </div>
      <div class="col-sm-3">
         <h3><span class="cc3">Redes sociais</span> / 18 horas atrás</h3>
         <p><a href="#">Facebook estreia botão "retweet" no leitor de notícias Paper</a></p>
      </div>
      <div class="col-sm-3">
         <h3><span class="cc4">Tecnologia</span> / 18 horas atrás</h3>
         <p><a href="#">AMD anuncia primeira CPU baseada em ARM de 64 bits</a></p>
      </div>
   <a href="#" class="mais-noticias">+ noticias</a>
      </div> <!-- /.container -->
   </section> <!-- /.row -->
</div> <!-- /.container-fluid -->
```

- **CSS**

```
1.  .noticias {
2.     position:relative;
3.     background:#333;
4.     padding: 10px 0;
5.     margin-top:20px;
6.  }
7.  .noticias h3 {
8.     font-size:90%;
9      color: #999;
10.    text-transform:uppercase;
11. }
```

```
12. .noticias a {
13.     display:block;
14.     font-size:150%;
15.     color:#fff;
16. }
17. .noticias a:hover { color:#ccc; }
18. a:hover { text-decoration:none; }
19. a.mais-noticias {
20.     width:160px;;
21.     font-size:100%;
22.     text-align:center;
23.     background:#666;
24.     position:absolute;
25.     left:50%;
26.     bottom:-10px;
27.     margin-left:-80px;
28. }
29. @media screen and (max-width:1200px) {
30.     body {
31.         background-image:none;
32.     }
33. }
34. @media screen and (max-width:992px) {
35.     form[role="search"] {
36.         float:none!important;
37.     }
38. }
```

[.../c6/imasters-final.html]

Código comentado:

Linha	Descrição
Linhas 1 a 6	Convém destacar a declaração de posicionamento relativo para a linha container desta seção. essa declaração visa a criar um contexto de posicionamento para o link +notícias (ver linhas 24 a 27).
Linhas 7 a 28	Estilização geral auto explicativa.
Linhas 29 a 38	Regras específicas para breakpoins com a finalidade de evitar pequenas quebras no layout. Retire estas regras e observe o efeito ao redimensionar a janela do navegador.

Observe na figura 6.8 o resultado da sexta etapa.
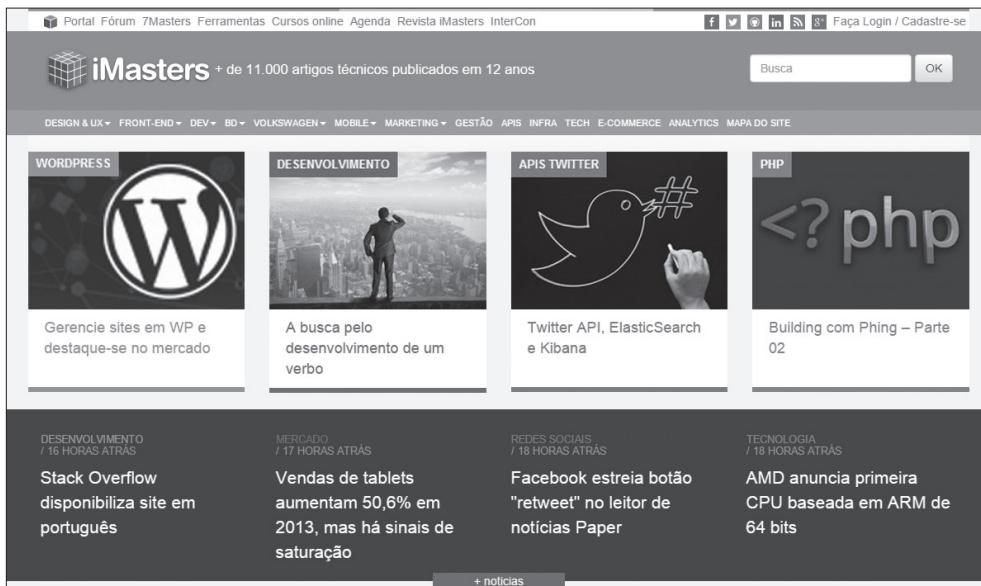
Figura 6.8 – Sexta etapa da criação do layout.

Consulte o arquivo desta etapa e visualize o resultado no navegador. Redimensione a janela do navegador e observe o comportamento do layout.

Referência

Site do Bootstrap. *http://getbootstrap.com*.